



HAL
open science

Auxiliary learning & Adversarial training for Medieval Manuscript Studies

Imad Eddine Ibrahim Bekkouch

► **To cite this version:**

Imad Eddine Ibrahim Bekkouch. Auxiliary learning & Adversarial training for Medieval Manuscript Studies. Musicology and performing arts. Sorbonne Université, 2024. English. NNT : 2024SORUL014 . tel-04555309

HAL Id: tel-04555309

<https://theses.hal.science/tel-04555309v1>

Submitted on 22 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE V (433) : Concepts et langages
Laboratoire de recherche Institut de recherche en Musicologie

THÈSE

pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ SORBONNE UNIVERSITÉ

Discipline : Informatique pour les sciences humaines et sociales

Présentée et soutenue par :

BEKKOUCH Imad Eddine Ibrahim

le : 11 janvier 2024

Auxiliary learning & Adversarial training pour les études des manuscrits médiévaux

Sous la direction de :

M. Frederic BILLIET – Professeur des universités, Sorbonne Université
Mme Victoria EYHARABIDE – Maître de conférences, Sorbonne Université

Membres du jury :

M. Frederic BILLIET – Professeur des universités, Sorbonne Université
Mme Victoria EYHARABIDE – Maître de conférences, Sorbonne Université
M. Attilio FIANDROTTI – Maître de conférences, Université de Turin
Mme Laurence LIKFORMAN-SULEM – Maître de conférences, Telecom ParisTech
M. Claude MONTACIÉ – Professeur des universités, Sorbonne Université
Mme Susan BOYNTON – Professeur, Columbia University

Keywords: auxiliary learning ; adversarial training ; knowledge graph embeddings ; neural network embeddings ; domain adaptation ; cultural heritage ; medieval manuscript studies

Mots clés : apprentissage auxiliaire ; réseaux antagonistes ; graphe de connaissances ; réseaux de neurones ; domain adaptation ; patrimoine culturel ; études des manuscrits médiévaux

This thesis has been prepared at

Institut de recherche en Musicologie

Maison de la Recherche, 28 rue Serpente, 75006
Paris, FRANCE

Web Site <https://www.iremuscnr.fr/>



*Je dédie à mes parents Saadi Souad et Bekkouch Boudjema, Mon frere Bekkouch
Seif et sa femme Herkati Sara, mes amis et mes collegues Valerie le Page, Adil
Khan, Youssef Yousseri, Benahmed Sofiane, Gérard BIAU, Nora ROGER,
Xavier Fresquet et toute l'équipe SCAI.*

À mon directeur Professeur Frederic Billiet

À ma co-directrice Victoria Eyharabide

Science is a way of thinking, not
just a body of knowledge.

Carl Sagan

The best way to predict the future
is to create it.

Peter Drucker

The science of today is the
technology of tomorrow.

Edward Teller

Contents

Contents	xv
List of Figures	xix
List of Tables	xxiii
Abbreviations	xxvii
Introduction	1
Context	1
Goals and research questions	3
Challenges	4
Data Scarcity & variation	4
Domain Drifts	4
Contributions	6
Annotated datasets	6
Publications	7
Outline	10
1 Related Works	11
1.1 AI for Medieval Manuscript Studies	11
1.2 Computer Vision	13
1.2.1 Image Classification	13
1.2.2 Object Detection	15
1.2.3 Few-Shot Object Detection	19
1.2.4 Image Segmentation	21
1.3 Generative Adversarial Networks	22
1.3.1 Generative models	22
1.3.2 Adversarial Learning	23
1.3.3 Transfer Learning	25
1.3.4 Domain Adaptation	25

1.4	Metadata & Knowledge Graphs	26
1.4.1	Knowledge Graphs	27
1.4.2	Graph Embeddings & creation	28
1.4.3	Embeddings	29
1.5	Conclusion	30
2	Datasets	31
2.1	Introduction	31
2.2	State of the art	32
2.2.1	Illuminated manuscripts datasets	33
2.2.2	Historical manuscripts datasets	33
2.3	Annotated MIMO (AMIMO)	34
2.3.1	The MIMO dataset	34
2.3.2	String instruments	36
2.3.3	Annotation Process	37
2.3.4	Class distribution	38
2.4	AnnVihuelas	40
2.5	AnnMusiconis	42
2.5.1	Types of Images	44
2.5.2	Meta Data	44
2.5.3	Classes	45
2.6	Medieval Musicological Studies Dataset (MMSD)	46
2.6.1	Image Dataset of Illuminations Representing Medieval Singing	46
2.6.2	Annotation of Written Supports in Illuminations	48
2.7	Conclusions	49
3	Dual Training for Transfer Learning	51
3.1	Introduction	51
3.2	Method	54
3.2.1	Baselines	55
3.2.2	Dual Training for Transfer Learning	56
3.3	Datasets & Challenges	58
3.4	Results	62
3.4.1	Global Performance Evaluation	62
3.4.2	Per-class Performance Evaluation	63
3.5	Conclusion	65
4	Few Shot Object Detection	67
4.1	Introduction	67
4.2	Methodology	70
4.3	Results	72

4.3.1	Global Few-Shot Object Detection Benchmark	74
4.3.2	Worst-Case Few-Shot Object Detection Benchmark	76
4.4	Conclusions	77
5	Auxiliary learning: Adversarial Domain Adaptation	79
5.1	Introduction	79
5.2	Related Work	83
5.2.1	Discriminator	83
5.2.2	Image Reconstruction	83
5.2.3	Pseudo Labeling	84
5.3	TripNet: Category Based Adversarial Domain Adaptation	84
5.3.1	Overview	84
5.3.2	Architecture	84
5.3.3	Losses	86
5.3.4	Optimization	88
5.4	AugNet: Augmentation Based Adversarial Domain Adaptation	89
5.4.1	Methodology	89
5.4.2	Architecture	89
5.4.3	Losses	91
5.4.4	Optimization	93
5.5	Experiments and Results	93
5.5.1	Toy Datasets and Results	93
5.5.2	Medieval Dataset Results	98
5.6	Conclusion	99
6	Auxiliary learning: Knowledge Graphs & Domain Adaptation	101
6.1	Introduction	101
6.2	Cultural heritage datasets	103
6.2.1	Medieval and Renaissance musical iconography as source and target domains	103
6.2.2	MusicKG: a knowledge graph of medieval musical iconography	105
6.3	Methodology	106
6.3.1	Architecture	107
6.3.2	Losses	109
6.3.3	Optimization	110
6.4	Results	112
6.4.1	Class level Evaluation	112
6.4.2	Target size Evaluation	113
6.5	Conclusion	114

7	Auxiliary learning: Domain Generalization	117
7.1	Introduction	117
7.2	Methods	121
7.2.1	Domain Generalization	122
7.3	Analysis	125
7.3.1	Benchmarking Datasets	126
7.3.2	Domain Generalization Results	126
7.3.3	Unsupervised Domain Adaptation	129
7.3.4	Over-fitting	131
7.4	Conclusion	134
8	Musiconis: Populating the database while advancing research	135
8.1	Project: Medieval Singing	136
8.2	Project: Medieval Musical Instruments	137
8.3	AI-powered search	137
8.4	Results and shortcomings	140
8.5	Continuous Improvement	142
9	Conclusion	145
9.1	Contributions	145
9.1.1	Datasets and benchmarks	145
9.1.2	Transfer learning techniques	146
9.1.3	Auxiliary learning techniques	147
9.1.4	Augmenting Musiconis	148
9.2	Perspectives for future work	149
9.2.1	Rules and clues	149
9.2.2	Leveraging high dimensional data	149
9.2.3	Image generation	150
9.2.4	Interpretability	151
	Bibliography	153

List of Figures

1	The Virgin and Child with Angels; Lieven van Lathem (Flemish, about 1430 - 1493); about 1471; Tempera colors, gold leaf, gold paint, silver paint, and ink on parchment; Leaf: 12.4 × 9.2 cm (4 7/8 × 3 5/8 in.); Ms. 37 (89.ML.35), fol. 5v; No Copyright - United States (http://rightsstatements.org/vocab/NoC-US/1.0/)	5
1.1	Cutting-edge generative model taxonomy.	24
1.2	The hierarchy of concepts (entities) in the MusicKG knowledge Graph.	28
2.1	An image segmentation example showing a female musician playing the vielle, female and male dancers from the AnnMusiconis database	32
2.2	Examples of chordophones in the MIMO database.	34
2.3	Zither-Harp - Made in Germany - 1942	35
2.4	Qanun - Türkiye - 1825	35
2.5	Harp 40 strings. 7 pedals. Hook mechanics. Paris 1783.	35
2.6	Violoncello - London - 1792	35
2.7	4 examples of images from the AMIMO dataset.	35
2.8	An image segmentation example showing an annotated French Harp from the 1790's in the MIMO dataset.	39
2.9	Angel playing vihuela - 13cent - vihuela de péñola - Salamanca, Catedral vieja, Capilla del Aceite	41
2.10	Angel musician with three-string vihuela or guitar. Lorenzo Mer- cadante de Breaña, 1464-1467	41
2.11	Viola da mano played by Serafino Aquilano	41
2.12	Angel guitarist from "La Presentación de Jesús en el Templo." painted by Diego Valentin Díaz. 1600-1650.	41
2.13	4 examples of images from the Vihuela dataset.	41
2.14	King David playing the vièle - 1151 - Troyes, Grand Est, France.	43
2.15	The Assumption of the Virgin with Saints Michael and Benedict - 1493 - Cortona, Toscane, Italie	43
2.16	Putto Playing the luth - 1540 - Auch, Occitanie, France	43

2.17 King David playing the harp inspired by Saint-Esprit - 1200 - Paris, France.	43
2.18 4 examples of images from the Muiconis dataset.	43
2.19 A screenshot from Musiconis website of a Shepherd holding a bag-pipe.	45
2.20 An annotated phylactery example	47
2.21 The average and standard deviation of critical parameters	50
3.1 An example of a Musiconis' image to show the small size of the musical instrument compared to the size of the entire page	53
3.2 Dual Training for Transfer Learning method. The images are extracted from MIMO dataset (Source) and Musiconis (Target).	55
3.3 Four examples of vielles. (a) Stone statue in Musiconis database. (b) Manuscript in Musiconis database. (c) Painting in Vihuelas database. (d) Photograph in MIMO database.	58
3.4 The kernel density estimation of empirical probability density function of the object area to image size ratios for MMV datasets. The sections represent the range we chose for deciding small, medium, large sizes for evaluation.	61
3.5 Comparison between results of our method DTTL (on bottom) vs traditional transfer learning method DKT (on top).	64
4.1 The average and standard deviation of critical parameters	69
4.2 Inference of the Mask RCNN architecture for few-shot instance segmentation on our medieval singing dataset.	75
5.1 Case study of domain adaptation. Domain adaptation aims at closing the discrepancy gap between different datasets from different domains SVHN, i.e., street view house numbers and MNIST, i.e., hand written digits, while preserving good performance on a specific task, i.e., digits classification.	80

5.2	The Architecture of our model. It can be divided into three parts: an encoder, a discriminator and a classifier. The encoder translates the images (i.e., X space) to embeddings in the latent space (i.e., Z space). In the latent space, each group of embeddings is marked by either S_i or M_i , where i is the label of the image, and the prefix letter notates whether it is from MNIST (M) or SVHN (S). Thus, the Z space can be expressed as $Z = Z^s \cup Z^t$, where $Z^s = \bigcup_i S_i$ and $Z^t = \bigcup_i M_i$. The latent representation is fed to both the discriminator and the classifier. The discriminator distinguishes if the latent representation is from source or target domain, whereas the classifier finds the suitable label for it.	85
5.3	This is a projection of the latent representation of both source domain (SVHN) and target domain (MNIST), to describe their distribution in the latent space, before and after domain adaptation. Each cluster is labeled by M- i or S- i to denote if it belongs to the MNIST or SVHN, respectively, and i represents the label associated with each cluster.	96
5.4	Comparison between TripNet and DupGAN in terms of number of epochs needed for convergence for SVHN \rightarrow MNIST case. This shows that the generative model comes with its high cost of training time.	97
6.1	An example of artwork in the AnnMusiconis database	104
6.2	Four examples of Renaissance musical instruments in the Vihuelas database	104
6.3	Representation of the artwork instance describing the example of Fig. (Eyharabide 2021) 6.1	105
6.4	An overview of the proposed approach	107
7.1	A horse wrongly predicted as an Arabian camel by ResNet, because of the surroundings. The left part is the LIME interpretation of the ResNet decision.	118
7.2	A horse wrongly predicted as a macaw parrot by ResNet, because of different colors (painting). The left part is the LIME interpretation of the ResNet decision.	119
7.3	Model Architecture: The Encoder generates latent representation z which is used by the Decoder to reconstruct the input using L_R and by the Classifier to classify the sample using L_C . The encoder is trained on the classification L_C and adversarial L_A losses.	121

7.4	Reconstructed images formed by training a decoder on a model (Encoder+Classifier) trained only for classification. Reconstructed on the left, Input image on the right.	122
7.5	Reconstructed images formed after applying our ARL Generalization and training a new decoder to reconstruct the input images. Reconstructed on the right, Input image on the left.	128
7.6	Comparison of different models on the task of digit classification on MNIST for the over-fitting scenario. The accuracy results are reported as the average of 5 experiments with the best hyper-parameters. OF is the over-fitted model, which is used by O-ARL as the initial start for solving the over-fitting problem. WT is the well trained model, T-ARL is the model which is trained from the start with ARL, and F-ARL-Sep is the WT model and fine-tuned with both ARL and sep loss7.4].	133
8.1	A stone romane sculpture of a Fiddle-playing musician from the 11th century from Saint-Martin-de-Boscherville, Normandie, France.	138
8.2	A stone romane painting of a Fiddle-playing musician from the 13th century from Retjons, Nouvelle-Aquitaine, France.	138
8.3	Two example images from Musiconis library of medieval musical artworks.	138
8.4	An example of singing in churches used to train the medieval signing search engine.	139
8.5	Object Detection Search Engine steps.	141
8.6	Model's results for medieval singing, the manuscript includes a book and lutrin and two singers with traditional religious clothes	143
8.7	Model's results for instrument search.	143
8.8	Two example images found by the instance segmentation search engine.	143
8.9	A simplified overview of the machine learning pipeline.	144
9.1	DALL-E mini by craiyon.com.	150
9.2	DALL-E by OpenAI.	150
9.3	Two example images generated by AI for the following sentence: "King David playing the harp in medieval manuscript"	150
9.4	Viele in a medieval manuscript.	151
9.5	King playing the harp in medieval manuscript.	151
9.6	Two example images generated by AI for the following sentence: "King David playing the harp in medieval manuscript"	151

List of Tables

2.1	The distribution of objects in AnnMusiconis across different classes.	39
2.2	Distribution of objects in our AnnMusiconis across different classes.	45
2.3	Distribution of objects in MMSD across different classes.	48
3.1	Per-class Performance map comparison between several Object Detection models on the task of Transfer Learning. TL refers to vanilla Transfer Learning, DKT refers to Dual Knowledge Transfer and DTTL refers to our method. The experiments use MIMO dataset as a source dataset, and either AnnMusiconis or AnnVihuelas as the target.	61
3.2	Per-class Performance map comparison between several Object Detection models on Transfer Learning’s tasks. TL refers to vanilla Transfer Learning, DKT refers to Dual Knowledge Transfer, and DTTL refers to our method. The experiments use the MIMO dataset as a source dataset and either AnnMusiconis or AnnVihuelas as the target.	62
3.3	Hyper-parameters for the transfer learning experiments on musical instruments object detection for Table 4.1 and Table 3.2. The annotations are described in Algo 2].	63
4.1	Average Precision evaluation of different state-of-the-art models for object detection using our newly proposed dataset. LB refers to the lower bound baseline, which is transfer learning with the same data, and UB refers to upper bound baseline, which is a transfer with the full dataset.	75
4.2	Average precision evaluation of different state-of-the-art models for object detection using our newly proposed dataset. LB refers to the lower bound baseline, which is transfer learning with the same data, and UB refers to the upper bound baseline, which is a transfer with the full dataset.	76

5.1	The test accuracy comparison for UDA on digit classification. The results for the previous works have been copied from the original papers or the DupGAN [67] without repeating the experiments because we used similar architecture for the encoder and the classifier part as well as the same experimental setup as those works. The "-" notation is used for experiments where the results have not been reported in previous works.	94
5.2	The best set of hyperparameters for TripNet for the experiments reported in Table 5.1. β_{Sep} , β_P and β_C are the balancing parameters for the triplet loss from equation 6.6. λ_T and λ_S are the balancing parameters between the source and target classification losses from equation 7.1. PL_{Thresh} is the minimum confidence level provided by the classifier so that the image would be considered in pseudo labeling.	95
5.3	The test accuracy comparison for UDA on digit classification.	97
5.4	Test accuracy for the ablation study for TripNet	98
5.5	The test accuracy comparison for UDA on Musical Instruments Recognition In Medieval Artworks.	99
6.1	Per class F1-score comparison between our model and three baselines.	113
6.2	Performance evaluation based on f1-score of KGE-DA method while varying target data sizes	114
7.1	Domain Generalization for digit classification: RMNIST. The average accuracy over 20 runs of the model. We represent each experiment by the name of its target dataset.	127
7.2	The test accuracy comparison for DG on Musical Instruments Recognition In Medieval Artworks.	128
7.3	Digit Recognition Benchmark on the MNSIT-USPS-SVHN dataset for Unsupervised Domain Adaptation. Each experiment name follows source domain - target domain naming convention. ARL-sep is used to reference to our method + the seperability loss and ARL is used to reference our model without it. The "-" notation is used for experiments where the results have not been reported in previous works.	129
7.4	Multi-source Unsupervised Domain Adaptation results on PACS datasets obtained as average over five runs for each experiment.	130
7.5	Accuracy results of different models on digit classification datasets MNIST-USPS-SVHN and MNISTR for the Over-fitting scenario. The best model is bolded and the second best is underlined.	131

7.6	Hyper-parameters for the over fitting experiments on digit classification Table 7.5. G-epochs is generalizing epochs and PT-epochs is pretraining-epochs.	132
-----	---	-----

Abbreviations

AI Artificial Intelligence
AMIMO Annotated Musical Instrument Museums Online
BnF French National Library
CH Cultural Heritage
CNN Convolutional Neural Networks
GAN Generative Adversarial Network
IIIF International Image Interoperability Framework
IoU Intersection Over Union
KG Knowledge Graph
mAP Mean Average Precision
MIMO Musical Instrument Museums Online
MMS Medieval Manuscript Studies
MMSD Medieval Musicological Studies Dataset
NMS Non Maximum Suppression
RCNN Region-based Convolutional Neural Networks
RPN Region Proposal Network
TL Transfer Learning
UDA Unsupervised Domain Adaptation
ViT Visual Transformers
Yolo You Only Look Once

Introduction

In this introduction, we describe Medieval Manuscript Studies (MMS) and some AI research fields that we believe can help this domain. This thesis is at the intersection of AI and MMS, aiming to showcase the utility of leveraging AI as a helper for Musicology experts. This thesis is at the intersection of AI and MMS, aiming to showcase the utility of leveraging AI as a helper for Musicology experts working in the field of MMS. We start by defining the scope and context of our thesis in a European setting. Then, we defined our research statement and goal and provided a description of the thesis outline. Finally, we finished by listing the contribution of this thesis to the fields of AI and MMS.

Context

Cultural Heritage is the legacy we inherited from our past generations and maintained in the present for the benefit of the generations to come. Preserving a nation's history is of high importance since it provides a unique opportunity to introduce its identity. Cultural heritage is rooted in the identity of the people as it reflects the values, hopes, and beliefs of a region and maintains the integrity and unity of the people. Musical Heritage is especially fragile and intangible, since safeguarding it and transferring it through generations is harder than protecting buildings and monuments. One major window to our musical history is manuscripts as they can provide clues on dance performances, songs, musical instruments used, types of clothing, and the different historical festivities. The BNF¹ and other major museums have launched a big digitization campaign of medieval manuscripts, with over 370,000 manuscripts. Museums also add metadata to the manuscripts which are well-detailed and structured descriptions such as the date and Title. Digitalization allows musicologists quick access to manuscripts and documents but with a large amount of content, searching for the right item becomes impossible.

As part of the European Commission's mission, preserving and reconstructing

¹<https://www.bnf.fr/en>

our past is a challenge of high importance. Projects such as the reconstruction of the Notre Dame de Paris Cathedral in Paris France made it very clear that our past is very fragile and preserving it is a mission that can not be postponed. The European Union is funding a multitude of projects in the context of preserving European cultural heritage. Some examples are the EU-funded RePAIR project which aims at facilitating the reconstruction process of bringing ancient and historical artworks back to life which can be one of the most labor-intensive steps of archaeological research. Other projects such as AI4Europeana aim at building an Artificial Intelligence platform for the cultural heritage data space by providing access to a large pool of AI resources such as labeled datasets and basic AI tools. Not only is preserving cultural heritage an important mission for the European Union, but it is also a goal for large companies such as Microsoft with their AI for Cultural Heritage initiatives. The aim of such initiatives is to provide more realistic experiences of ancient worlds and uncover art histories through AI and knowledge graphs. Closer to our university, a project of immense importance to the field of cultural heritage preservation is project PHEND ² (The Past Has Ears at Notre Dame (2020-2024)) which is a French Collaborative research project founded by the ANR aiming at better understanding the sonic history of the Notre-Dame cathedral in pairs.

Computer vision models are able to provide the human-level performance of several tasks such as classification, object detection, and instance segmentation due to the large volumes of images online and the computing powers of GPUs. Yet their performance degrades drastically when applied to historical data due to the scarcity of data, large variations in style, and small size of interest objects in the images. This problem is commonly known as a domain gap and it is closely related to over-fitting. Researchers [68] have developed several techniques to deal with such a problem depending on the context task and they all fall under the transfer learning umbrella [47].

Auxiliary learning [101] is a research field that aims at improving the results of classification models by leveraging and designing auxiliary tasks that can improve the performance of the primary task. The underlying assumption is that learning with an auxiliary task can improve the ability of the model to generalize to unseen data. This allows it to not over-fit on the training of the primary task such that the auxiliary task can be a distraction or a regularization that hinders the abilities and flexibility of the model. This sadly comes at a great cost, which is re-labeling the data manually to provide the auxiliary labels for the training. The sharing of the learned weights between the auxiliary model and the primary model results in the extraction of much more abstract and rich high-level information. This helps in discriminating between the different classes which otherwise would not have

²<http://phend.pasthasears.eu/>

been learned from the primary task training only. This is similar to multi-task learning which proved to be an extremely effective method of training with one major difference which is that the only performance that matters is the primary task. With supervised auxiliary learning can be manually chosen to complement the primary task based on domain knowledge and the ability of the re-labeling. Unsupervised Auxiliary learning on the other hand removes the requirement of manual labor and domain-specific tuning, but that drastically reduces the increase in performance resulting from using the techniques. In this thesis, we will cover multiple methods that use both supervised auxiliary learning (as we have multiple labels for each image thanks to the annotations of the museums that digitalize the manuscripts such as the date, location, type of supporting materials, artists, some content specific annotations etc) and unsupervised auxiliary learning.

Goals and research questions

The goal of the thesis is to explore different ways that we can leverage AI for MMS. We focus intentionally only on computer vision tasks as it is more applicable to the field of MMS, especially Medieval Musical Iconography which focuses on the study of music depictions in the manuscripts. It offers us information about performers, musical instruments, and practices of the Middle Ages (a.d. 500 to about 1500). We summarize our research statements in the following questions:

- How can we use computer vision-powered AI to search for objects and patterns of interest in the vast amount of digitalized data? : After collaborating with multiple musicologists working inside the IReMus laboratory, the most common problem they are facing is the time-consuming and repetitive task of searching manually in museums websites for images of a specific instrument or a pattern of singing.
- Is it possible to use images from other domains outside of the field of MMS to improve the performances of models on MMS tasks? : Immediately after considering the first question mentioned earlier we are faced with a big problem which is lack of data in the field. So our next logical step is to get more data from outside the field sharing similarities with medieval data.
- How much data is needed to train a model for MMS computer vision tasks effectively? : depending on the difficulty and details required for the MMS research that musicologists are interested in, we might be able to start with a large amount of data or with a very small dataset. The question that arises is, how much data they need to have in order to leverage computer vision AI as a search engine effectively.

- Is it possible to use other data types to improve computer vision models' performances in the field of MMS? : The most creative part of this thesis aims at solving the problem of lack of image data by using other types of information commonly available in the field of MMS since these manuscripts come with detailed metadata in multiple forms.

Challenges

The field of Medieval Manuscript Studies doesn't have enough active AI research and the possibilities are endless. This led us to have multiple options for exploring and innovating in the thesis, but also multiple challenges ahead. Namely, we have two major issues:

Data Scarcity & variation

The first issue we faced when we wanted to apply computer vision models for both classification and object detection was the lack of annotated datasets in the field. Since there are many types of objects to annotate (singers, musical instruments from multiple centuries and countries, sculptures, stained glass, etc), this makes it impossible to find a dataset tailored to the musicologist's interest. But even after annotating the few images a musicologist probably has for their object of interest, it is not enough for a computer vision model, as the shapes and colors and styles of that same object change drastically from one source to the other and even in the same museum, same country and same century we can find many differences due to the artist's expression and style. This makes it very challenging to work with MMS datasets and lead us to annotate multiple types and variations of datasets and to explore the fields of transfer learning and few-shot object detection.

Our interest in medieval and historical data makes it very challenging to get images of the objects that we are interested in finding mainly because research questions tend to concern objects that are too specific and hence cannot be included in the metadata. Figure [1](#)³ shows clearly a medieval manuscript that contains 6 musical instruments that are completely ignored by the museum's annotators who focus only on the core of the image which is The virgin and child with angels.

Domain Drifts

Domain drift is a major problem for machine learning models as is still an active area of research. The issue arrives when the model is trained on a dataset of a

³<https://www.getty.edu/art/collection/object/105T01>



Figure 1: The Virgin and Child with Angels; Lieven van Lathem (Flemish, about 1430 - 1493); about 1471; Tempera colors, gold leaf, gold paint, silver paint, and ink on parchment; Leaf: 12.4 × 9.2 cm (4 7/8 × 3 5/8 in.); Ms. 37 (89.ML.35), fol. 5v; No Copyright - United States (<http://rightsstatements.org/vocab/NoC-US/1.0/>)

specific source and then tested and used on a dataset from a slightly different source. Even though the differences might not be visible for a human, the domain drift problem can drop the performance of the model drastically. In our case, the domain drift problem is extremely visible and reduces the performances of the models of MMS to unusable levels. The solution to this issue is the field of research known as Domain adaptation, which we explore in detail through various approaches of auxiliary learning and adversarial training.

Contributions

We split our contributions into two parts: the annotated datasets and the publications.

Annotated datasets

Our initial contribution is the creation of four new annotated datasets (of museum images) in the field of Medieval Manuscript Studies focused on different tasks relating to medieval singing and instruments for the tasks of object detection and instance segmentation:

- Annotated Musical Instrument Museums Online (AMIMO): The first dataset of images we extracted from Musical Instrument Museums Online (MIMO)⁴ with 10258 manually annotated images. AMIMO focused on real photographers of medieval and historical string instruments.
- AnnVihuelas: The smallest dataset we created from Vihuelas⁵ with only 165 images of the stringed instrument Vihuela, which was popular in Spain during the Renaissance.
- AnnMusiconis: Musiconis⁶ is an image archive for musical performances of the medieval period. We annotated 662 chordophones.
- MMSD: Medieval Musicological Studies Dataset (MMSD) is a dataset of 693 objects mainly focused on books and lecterns and alters as indices to search for medieval singing practices portrayed in medieval manuscripts.

These datasets allowed us to create and innovate different experimental setups and solutions for the challenges we faced while applying AI on MMS.

⁴<https://mimo-international.com/MIMO/instrument-families.aspx>

⁵<https://vihuelagriffiths.com/>

⁶<https://musiconis.huma-num.fr/fr/>

Publications

Our first AI contribution was a new black box method for object detection models that satisfies the constraints we face in the field of medieval studies, such as being reproducible non-intrusive and model-independent. Our method is called dual training for transfer learning and it leverages three datasets, an original unrelated (used for lower-level feature extraction) a mid-size related dataset (similar classes but different styles) and the target dataset. Our method trains the model in an iterative manner between related and target datasets in order to improve the final result over the target data. The experimental trials we performed affirm that our technique outperforms vanilla Transfer Learning with +8.83% F1-score on our medieval datasets.

We also tested and extended our transfer learning method to work in situations where data is extremely scarce. This leads us to explore the field of few-shot object detection by proposing a new and simple method for black box few-shot object detection, that works with all the current state-of-the-art object detection models such as Yolo, RCNN family, and Visual Transformers.

Black box methods [96] are quite easy to generalize and provide an increase in performance across the board for multiple types of models. But, the effects noticed by black-box methods are not significant enough to build powerful machine learning models starting on a dataset that suffers from lack of data, high resolution (5-10 times higher resolution than common images used to train large-scale models), the small size of the object of interest, deformed images, and a major variety of style and underlying support material. All of these problems make it almost impossible to build an acceptable model directly without making significant changes to the model's architecture and training algorithm. Sadly, creating a new architecture requires abandoning all of the knowledge contained within the weights of large pre-trained models which makes it an unfeasible solution. Hence, we decided to leverage the current architectures by making some improvements to the models and the training steps. The method we chose to use is Auxiliary learning, which is more invasive and more timely but yields much better results.

Moving to the most important contributions we made, which are the domain adaptation methods we proposed based on adversarial learning and auxiliary learning. Our contributions are considered part of the self-supervised learning and unsupervised auxiliary learning tasks, which are highly related to domain adaptation. We presented multiple techniques of domain adaptation that leverage Adversarial Learning and Knowledge Graph embedding.

The third method is a new method for Unsupervised Domain Adaptation that is at the same time fast and resilient, which we split into two sections (TripNet and AugNet). Taking as input pictures, the model tries to learn a good classifier and maintain a neutral and unbiased encoder. The original idea was to take two

separate data sources and build a source-detector/classifier that we use to train the encoder in an adversarial manner. This TripNet idea works great but we wanted to push it even further for cases where we don't have the metadata related to the source of the images we are using, so we assume the data is all the same. AugNet aims to solve this issue by leveraging augmentations to transform the images into new domains where each augmentation is considered a new domain. The new detector is no longer trying to predict the source of the image but trying to predict the type of augmentations the image received. This makes it a lot more inclusive and applicable but it adds a layer of complexity notably for classes that get destroyed after some specific augmentations (like a 6 becoming a 9 after rotation, etc). The previous two methods assume the least amount of assumptions on the training data, aiming to be applied to images from all domains. However, our thesis is interested in medieval manuscripts, which tend to be available in museum libraries along with a very detailed descriptive meta-data. Hence we decided to leverage this idea by putting this meta-data information into a graph and using Node2Vec to transform these nodes into anchors that clean up and organize the latent space from the encoder, and hence providing more information and separability that the classifier can leverage to get unbiased and accurate classifications.

Our final contribution is the application of a Domain generalization technique that assumes zero information about the input images and treats the image pixels themselves as a source of randomness and noise that hurt the classifier. Hence, it adds a decoder on top of the classical classification architecture instead of a detector which aims to regenerate the input image with all its background noise while the encoder is trained on the adversarial loss of the reconstruction in order to forget the noise and style information while focusing on the classification part of the training.

We summarized our published articles for the thesis in the following list:

- I. E. I. Bekkouch, V. Eyharabide and F. Billiet, "Dual Training for Transfer Learning: Application on Medieval Studies," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: <https://doi.org/10.1109/IJCNN52387.2021.9534426> [14]. In this article, we presented our Transfer Learning method applied to Medieval Manuscript datasets.
- I. E. I. Bekkouch, V. Eyharabide, V. Le Page, and Frédéric Billiet. 2022. "Few-Shot Object Detection: Application to Medieval Musicological Studies" Journal of Imaging 8, no. 2: 18. <https://doi.org/10.3390/jimaging8020018> [72]. In this article, we presented our Few-shot object detection technique applied to medieval musicological studies and discussed the question of how much data is needed to build an object detection model.

- I. E. I. Bekkouch, N.D. Constantin, V. Eyharabide, F. Billiet (2022). Adversarial Domain Adaptation for Medieval Instrument Recognition. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems, vol 295. Springer, Cham. https://doi.org/10.1007/978-3-030-82196-8_50 [13]. In this article, we presented our first intrusive method for unsupervised domain adaptation, where we build a model using unlabeled inference data.
- V. Eyharabide, I. E. I. Bekkouch, and N. D. Constantin. 2021. "Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition" Computers 10, no. 8: 94. <https://doi.org/10.3390/computers10080094> [41]. In this article, we leveraged the meta-data available with our datasets to guide the computer vision models using a knowledge graph-based domain adaptation approach.

We also applied published algorithms to the domain of medieval manuscript studies from the following articles:

- I. E. I. Bekkouch, Y. Youssry, R. Gafarov, A. Khan, and A. M. Khattak. 2019. "Triplet Loss Network for Unsupervised Domain Adaptation" Algorithms 12, no. 5: 96. <https://doi.org/10.3390/a12050096> [11]. In this article, I presented the idea of removing generative models from domain adaptation techniques, which leads to faster and more generalizable models. This technique was tested on medieval manuscript datasets in Chapter 5.
- I. E. I. Bekkouch, D. N. Constantin, A. Khan, S. M. A. Kazmi, A. M. Khattak and B. Ibragimov, "Adversarial Reconstruction Loss for Domain Generalization," in IEEE Access, vol. 9, pp. 42424-42437, 2021, doi: <https://doi.org/10.1109/ACCESS.2021.3066041> [8]. In this article, I presented the domain generalization idea of leveraging the pixel values directly for an auxiliary task used to constrain the model's overfitting abilities.

Although I aimed to make the thesis a theoretical thesis with a focus on novelty in training algorithms, loss functions, and architecture, the final goal of the thesis is to provide much-needed AI-powered aid to musicology research. We did this throughout the thesis by collaborating with musicology experts mainly by providing them with new manuscripts that either confirm or reject their initial hypotheses allowing them to build a better understanding of medieval practices, especially around singing. Although every thesis chapter is centered around AI for Musicology, we made the last chapter a recap of the impact that our work had on musicologists by increasing the Musiconis library of medieval artworks, adding new musical instruments from the medieval periods, and multiple examples of medieval singing with a focus on religious singing inside of churches.

Outline

The remainder of this thesis is organized as follows:

- Chapter 1 is dedicated to exploring the state of the art in the field of Medieval Manuscript Studies and gives definitions to the concepts discussed in the thesis.
- Chapter 2 describes the datasets we annotated in detail and provides statistics about the classes and sizes of each dataset.
- Chapter 3 describes our Dual Training for Transfer Learning technique for object detection and provides empirical evidence for its efficacy.
- Chapter 4 is about the few-shot object detection technique and how we used it in the field of Medieval signing along with a detailed benchmark to evaluate the question of how many samples we need to get decent results for object detection.
- Chapter 5 focuses on adversarial domain adaptation techniques and shows cases two novel methods based on pseudo labeling, separability losses, and, most importantly, adversarial training. It proves that there is no need to use the generative part of the GAN as it is very time-consuming and doesn't provide that much better results.
- Chapter 6 goes deeper into the domain adaptation field by combining it with knowledge graph embeddings and leveraging the full range of metadata accompanying the museums' data that we typically work within the field of MMS.
- In chapter 7, we describe the final innovation of the thesis which is a domain generalization technique that reduces the amount of required metadata needed by simply treating the pixels themselves as an extra source of information leaning closer to the unsupervised side of auxiliary learning.
- Chapter 8, is the empirical evidence of our work and puts our methodology to use to help musicologists in the field of MMS. It starts by presenting Musiconis archive and showing two projects we worked on throughout the thesis: medieval singing and musical instrument recognition.
- Finally, we present a summary in chapter 9 that concludes the thesis contributions and presents a few perspectives for future research that can be based on this thesis.

Chapter 1

Related Works

Before going deeper into the methods and techniques used in this thesis, we start by discussing some AI-based innovations in MMS. Then, we discuss some computer vision methods, and we finish this chapter by discussing Generative Adversarial networks and knowledge graphs, as they are essential for our contributions.

1.1 AI for Medieval Manuscript Studies

In recent years, multiple deep learning applications applied to cultural heritage (CH) have been developed, especially for images of ancient paintings and historical artworks. Most of CH's applications are in the domain of computer vision. One of the major drawbacks of CH applications' implementation is the quality and quantity of annotated datasets available to train and test deep learning algorithms. In general, the data is scarce, and copyrights restrict their reuse and publication. Several historical manuscript image datasets have been proposed [120, 147, 143] with the goal of training and evaluating deep learning methods. Digitalized manuscripts datasets are mainly used for document analysis and text recognition [69, 119, 159]. Since we are interested in ancient musical instruments recognition, we need manuscripts that are illuminated. Among the existing illuminated manuscripts datasets, we can mention the HBA corpus [109], or the HORAE dataset [19], but none of them contain musical instruments. Other datasets containing artistic artworks are PrintArt [25], BAM [170], OmniArt [150].

Neural networks, ranging from Convolutional Neural Networks (CNNs) to Mask R-CNN [63], are useful in recognizing high-level artworks features from the low-level image features like colors, shapes, and texture. Hence, they are widely used in CH's applications manipulating images of artworks and paintings, such as object detection [59, 77], image classification [25, 110, 26], image description and captioning [162, 144, 27], or answering visual questions [51] in artworks and paintings.

Writer Identification is another field that has attracted attention in this area, which is classifying a page of handwritten historical document scans to their original author or artist. There exist two main streams for solving this task: the easier one is building a row-level model followed by a majority classifier using CNNs [33], whereas the other is a facial recognition inspired method which embeds the whole scan and computes distances between the training data and the new samples. Another promising application of deep learning to medieval and historical works is predicting the creating date of artworks, like the approach presented in [60] that used Convolutional Neural Networks and outperformed the traditional rule-based systems used before.

However, the use of deep learning techniques is not always enough to interpret artworks. The vision of a human expert is essential to understand the content and meaning of artworks fully. Therefore, state-of-the-art methods include ontologies and knowledge graphs that model human knowledge to improve neural networks results. These approaches with graph embeddings [52] and graph neural networks [173] aim at creating meaningful vector representations including structural graph information (such as nodes and edges) as well as the content information (such as texts or images) of each node. The method presented in this article [173] combining knowledge graph embeddings with visual embeddings is a clear example of such approaches.

Although the field of object detection is relatively mature and has been around for quite some time, its applications to cultural heritage data have been relatively modest. In the musicology field, most contributions use simple images [42], such as the recent contributions to digital cultural heritage analysis focusing on similarity metric learning methods for making semantic-level judgments, such as predicting a painting's style, genre, and artist [40, 176]. Other contributions detect fake artworks through stroke analysis and an artistic style transfer using adversarial networks to regularize the generation of stylized images [134].

Other applications of deep networks to archaeological remote sensing include topics such as the detection of buried sites on Arc GIS data [142] and the classification of sub-surface sites using R-CNNs on LiDAR data [86]. Both contributions followed a transfer learning approach by fine-tuning a pre-trained CNN using LiDAR data in ImageNet [11]. Overall, we can see that the application of computer vision in the digital humanities and cultural heritage is a field that is still being uncovered, mainly because of the lack of data, and this why is our method for few-shot object detection will open a door towards more contributions in the field, overcoming the barrier of the lack of data. It is important to note that the lack of data we are addressing, isn't the lack of well-documented publicly available images. The issue is with the lack of annotated data that can be used to train machine learning models. Examples of such image repositories for medieval manuscripts

are:

- **MMMO Database:** Medieval Music Manuscripts Online Database ¹ is an international virtual library of music sources using manuscripts of western language from the medieval period.
- **CVMA Database:** Corpus Vitrearum Medii Aevi ² is an international research project aiming to record medieval stained glass and provides free access to more than 28000 images³.
- **DIAMM Archive:** the Digital Image Archive of Medieval Music ⁴ presents images and metadata for thousands of manuscripts and publications.

1.2 Computer Vision

Medieval Manuscripts come in different shapes and forms, but the majority of them are treated as images or text corpuses. For our work, we focused mainly on the image source as it is the most common format of medieval manuscripts and the researchers we are collaborating with (musicologists) focus on the iconography part of the manuscript.

There are three main sub-tasks used for processing such medieval manuscripts, which are Image classification, Object detection, and instance segmentation. Depending on what the goal of the study is, we might need to work with different computer vision models. In this section, we will discuss in depth the works done in each sub-task, the technologies commonly used and their usage and possible applications to the field of medieval manuscript studies.

1.2.1 Image Classification

Traditional Machine Learning Methods

Image classification is one of the most well-researched regions of computer vision and is still a growing field. Automating the discrimination of different images into categories has always been a task of interest to researchers and industries alike. The first technologies used in the field are manual techniques that are not even learning-based but handwritten heuristics used to compare or extract important features from images. First methods used for image processing used techniques such as thresh-holding and edge detection (Canny Edge Detector [177], Laplacian

¹<http://musmed.eu/>

²<https://www.cvma.ac.uk/index.html>

³<https://www.cvma.ac.uk/jsp/index.jsp>

⁴<https://www.diamm.ac.uk/>

Of Gaussian [31], Sobel-Scharr-Prewitt-Roberts filters [131]) for pre-processing data and using Erosion and Dilation for improving the quality of the data as it is very common for medieval data to be damaged or missing some parts. The next steps are usually building Bags of Visual Words to classify the different images based on their textures, colors, sizes, and shapes. We start first by extracting the important features of each image by using a features extractor (HOG, SIFT, Harris Detector, Shi-Tomasi Detector).

The second step is Learn a visual vocabulary from the different features of all images by selecting the main features of our images, this step is usually done by k-means clustering where we cluster the feature vectors obtained from the previous step, the resulting cluster centers (i.e., centroids) are treated as our dictionary of visual words. The third step is to quantize each given arbitrary image, and quantify and abstractly represent that image using bag of visual words model. This is done by computing the nearest neighbor for the important features of the image with the features from our dictionary, usually done using euclidean distance, and taking the set of nearest neighbor labels to build a histogram of features presence. And lastly, we train a Support Vector Machine model to classify the extracted histograms into the classes of interest.

Deep Learning Methods

Ever since neural networks were invented, they faced one major problem which is data, because unlike machine learning models which make assumptions on the data and try to build a model that maximizes the likelihood of the data following its patterns, deep learning models make very few assumptions and are quite generic, but they require a lot more data to train effectively. The first deep learning model was the multi layer perception which takes represents the model as a hierarchical function mapper with its basic function named the neuron which applies a linear combination of its input followed by a non linear activation function to increase the complexity of the learned decision boundary.

MLP are feed forward neural networks [75] which are trained by back-propagation [89] and gradient descent and are thought to be one of the most generic learners possible, until the invention of attention based learning [161]. MLP although quite powerful for structured data and tabular data specifically they had a big problem catching up to machine learning models which required far less training time and resources until the adoption of convolutional operation into the field of deep learning. CNNs made a very clear and simple assumption that at the time was logical and not constraining which is that decision making for every pixel is only related to its closest neighbours and not the rest of the image. This assumption was quite correct and provided uncomparable results with machine learning models and the time of training, amount of data needed was drastically reduced

compared to MLP.

Current advancements in computer vision lead to believe that both MLP and CNNs are not the optimal solution for automating vision especially for large scale application. New research shows that more generic deep learning models can provide much better results by leveraging the excessive amount of unlabeled data and the big computational resources that big AI companies have [74]. Examples of such generic models can be attention based models which have dominated the field of Natural Language Processing for the last 2 years. We will cover attention based models in a following subsection in detail, but they provide a great advantage in general especially for our type of data since they appear to perform much better than other models on higher dimensional data and on objects of smaller size which are the most common types of pictures and objects that we work with in the field of culturale heritage and medieval data. One major problem with such models is that they require a lot of training, fine tuning and huge amounts of data to provide good results for the moment but we explore them in the rest of the thesis and examine good measures and techniques for transfer learning that can help with such drawbacks.

1.2.2 Object Detection

Object detection is one of the main research areas now that image classification has been well studied and explored. It also started way back as early as image classification models, with methods such as Viola Jones (VJ) [166], Histogram of oriented Gradients (HOG) [35] which are still used for tasks such as facial detection in mobile applications like snapchat and tiktok because of their fast and accurate performance on edge devices that doesnt have big computational powers.

Object detection [97] is the new major research field in computer vision thanks to its large applicaton areas and the need of industrial models that provide better and more resilient performance across large fields and application areas. We can split Object detection methods into two large categories, the one stage detector and the two stage detectors. In general, one stage detectors are faster but less accurate models whereas the two stage models are more accurate but take more time. These two categories have been largely dominated by two families of models which are the RCNN family and the YOLO[123] family. RCNN [16] stands for Region based Convolutional Neural Networks, which as the name suggests, it applies a Convolutional neural network to different regions of the image and tries to predict whether an object is present in that region.

Region based Convolutional Neural Networks

The first version of the model of the RCNN family was named "Rich feature hierarchies for accurate object detection and semantic segmentation" [57] and built a pretrained CNN model on imagenet and fine-tuned it on the domain specific images (warped proposal windows) used to extract features from different regions and sent for classification using a class specific Support Vector Machine classifier. The main difference between this architecture and a traditional classifier is the category-independent region proposal step at the start which includes different methods such as: objectness, selective search, category-independent object proposals, constrained parametric min-cuts (CPMC), multi-scale combinatorial grouping.

All of these previously mentioned methods aim at grouping different pixels of the image in regions that can be potential objects regardless of the type of the object (size, shape, category, rotation, ratio ...) and they do that by calculating different similarity metrics such as: color similarity, texture similarity, size similarity, and shape compatibility which are combined in a final metric named final similarity. The results of this model are very promising but they take so much time to train and make inference on, mainly because the selective search algorithm proposes around 2000 regions for the same image, and hence the training time becomes 2000 times bigger than that of a classification task. The same thing is noticed for inference where it takes up to 47s to make inference on a single image on a powerful machine, which are splitted into in general 2s for the selective search algorithm and 45s-46s for running the CNN on every single region.

Fast RCNN

The next version of the model's family [56] came on 2015 with the most interesting idea in the field and the most revolutionizing which was later included in more models. The main idea was to send the whole image once over the CNN and then do all the previous steps but not on the image itself but over the latent space. This leads to a problem which is that selective search provides objects proposals of different sizes which leads to different size regions in the latent space that will be later sent through a Fully Connected Neural Network which requires a fixed input size. The authors solved this problem with the introduction of a new Pooling layer which is still currently used in most of the state of the art object detection models and instance segmentation.

ROI pooling is a new method of pooling named after the application areas where it is applied, Region Of Interest pooling. Similarly to other pooling layers, the ROI pooler has no weights and hence doesn't slow down the training or reduce the generability of the model. The ROI pooling layer applies a similar technique to

max pooling which aims at converting the size and shape of the extracted features maps of a certain proposed region into a smaller size feature map which retains the most important feature activations of every kernel in the previous layer. Unlike max pooling, the output of the ROI pooler is independent of the size of its input because the filter sizes are calculated on the fly for each region to correctly map its dimensions to the chosen output size. The functioning of the ROI max pooling aims at splitting a hw ROI feature map into a predefined HW matrix of sub-windows of approximately similar sizes $h/Hw/W$ (in cases where h is not divisible by H , we take more values for the first parts of the grids) and then max-pooling the values in each sub-window into the corresponding output grid pixel.

Evaluation methods for object detection models

Object detection models have improved drastically over the years and they have made their way into many application areas such as self-driving cars and medical applications. Hence, validating and evaluating such models correctly is a task of utmost importance. Summarizing a model's performance in one numeric value isn't ideal and will always be problematic and not enough, since the majority of these metrics will not take into account minorities and special cases, but at least in the following pages we will provide a list of potential metrics that can help provide an objective estimation of how good or bad your object detection model is.

The majority of the metrics we will be using to evaluate the object detection models are actually built to evaluate classification tasks. Hence there is a need to convert an object detection task into a classification task, and for that we use Intersection Over Union. Intersection over union (IoU) [62] is a simple yet effective method that allows to compare the similarity of two bounding boxes, with 1.0 being the highest value possible for an absolute match and 0.0 being the worst case possible being a complete miss-match. We can calculate the intersection over union simply by dividing the area of the intersection of two boxes by the their union [1.1]

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{B \cap G}{B \cup G}, \quad (1.1)$$

where B is the predicted bounding box, G is the ground truth bounding box, and \cap and \cup represent the intersection and union operations, respectively.

Object detection in essence is the task of detecting the presence of an object in an image, hence if we take a threshold for IoU between the real boxes and the predicted boxes we can convert the object detection task into a binary classification task, with:

- True Positives: predicted box is similar to real box and they share the same class

- False Positives: predicted box doesn't overlap any real box of the same class, meaning an object that is predicted in the correct location but the incorrect label will be treated as a double negative value.
- False Negative: meaning there is a real box in the test set that have no overlapping box of the same class.
- True Negatives: we don't use the concept of True negatives since this will be a really high number compared to the rest even for a bad model.

Now that we know how to convert the object detection problem into a binary classification problem we can start building metrics on top of that. Here is a list of the most commonly used metrics to benchmark object detection models and architectures:

- Precision Recall: precision and recall are highly important metrics in the field of machine learning especially for difficult datasets. Precision is a ratio of true positives by all the positive predictions, which we want to maximize in cases of life and death situations or large decision making but we don't care about it that much in the case of search applications. whereas on the other side Recall is the ratio of true positives over the total real positive instances which we want to maximize the most in our field especially in the search applications.
- F1-score: is the harmonic mean between precision and recall, and it provides a balance or a trade off between the two.
- Box AP: in other words Box Average Precision, which estimates the performances of the model based on a combination of the precisions and recalls of the model at several confidence thresholds. It is similar to the concept of Area Under the Curve for classification tasks. It's values range from 0 to 1 (or 100%) and the higher it is the better.
- AP50, AP75, AP: which stands for Average precision, and has three major variations based on the method used for calculation. AP is the primary metric used for benchmarking and it stands for AP at IoU=0.50 to IoU=0.95 with incremental increases of 0.05. AP50 (and AP75 respectively) is its variations with IoU=0.50 (and IoU=0.75 respectively).
- AP Across Scales: another important set of variations of the AP metrics are the scale variations with APs, APm, APl being Average Precision Small (Medium and Large respectively). You can define your own small medium and large threshold but the ones that the literature refers to by default are

for small objects: $area < 32^2$ and medium with $32^2 < area < 96^2$ and large objects with $96^2 < area$. it is important to note that we are speaking in terms of pixels with the average model taking an image of $size = 224^2$ hence the small objects are less than 2% of the image size, medium between 2% and 18% and large is anything that is bigger.

Non Maximum Suppression

NMS (Non-Maximum Suppression) [164] is a post-processing step used in all object detection models to reduce duplicate detections of the same object due to overlapping anchors being used on the same location. The main idea behind the greedy NMS algorithm is to remove all predicted bounding boxes with high IoU overlap with high confidence predictions. NMS helps in removing duplicate detections and results in a cleaner output. However, choosing the right threshold for NMS is also essential and it should be done at an individual dataset level and is considered as a major part of the hyper-parameter tuning of the model.

1.2.3 Few-Shot Object Detection

Object detection, as a subtask of computer vision, has been the focus of tremendous research interest over the past few years, from traditional computer vision algorithms such as Viola Jones, and the Histogram of Oriented Gradients Detector, which are still commonly used in mobile applications for their speed and accuracy, to new deep-learning-based models [178, 130, 71] such as Yolo [123], RCNN [57], SSD [32], and others. We can split the deep learning-based object detection models into two subgroups: one-stage detectors and two-stage detectors. One-stage detectors are famous for their speed, which started with the Yolo Tree family (from v1, v2, v3, which are the original models up to v4, v5, and pp, which are extensions provided by separate researchers). Other one-stage detectors were introduced in the field, such as the Single-Shot MultiBox Detector (SSD), and RetinaNet [2]. The Single-Shot MultiBox Detector (SSD) introduced the multi-reference and multi-resolution detection techniques, allowing better accuracy. The developers of RetinaNet argued that the single-shot detectors have low accuracy because of the imbalance between background and foreground classes and introduced the focal loss so that the detector would focus more on challenging examples throughout the training phase.

The second category of models is two-stage detectors, which are known for their accuracy but lack speed compared with single-shot detectors. Two-stage classifiers are very useful in cases where the inference time is not crucial, and there is no need for fast or real-time processing. Such cases are widespread in cultural heritage studies or medical applications, where accuracy is more important than increasing

speed by a few microseconds. The branch of two-stage detectors started with the introduction of region-based convolutional neural networks (RCNNs), which combined deep learning with the traditional selective search algorithm, and which were later abandoned due to the development of faster RCNN models proposing a fully deep approach based on region proposal networks. Although RCNN-based models have dominated this side of the family tree of object detection, especially with the introduction of feature pyramid networks, other models have also been proposed for two-stage detectors, such as spatial pyramid pooling networks (SPPNet) [66], which demonstrated many ideas that later found their way into the RCNN family.

All the previously mentioned deep models are convolution-based, mainly because CNNs have dominated the field of computer vision due to their performance as of 2021. Lately, however, a new branch of models is being added to the computer vision field, which are attention-based models, rather than CNN-based models. Attention is an idea that has been dominating the field of natural language processing for a long time now, with models such as bert and GPT, which provide a human-like level of understanding of text and responding to questions. This trend has found its way to computer vision thanks to the paper “An Image is Worth 16×16 Words”, demonstrating an approach commonly known as vision transformer (ViT) [171], as well as its follow-ups which applied a transformer architecture on 16×16 non-overlapping medium-sized image patches for image classification. Although ViT provided a good speed-accuracy trade-off compared with CNN-based models, its successful application required a large-scale dataset and training, which was later fixed using data-efficient image transformers (DeiT) [155], which proposed several training algorithms allowing the vision transformers to be applied on smaller datasets. Vision transformers are aiming to replace CNNs and outperform them in terms of speed and accuracy, and the best example of this is the current state-of-the-art model for image classification and object detection/instance segmentation, the Swin Transformer [102], which builds a hierarchical vision transformer using shifted windows that can be used as a generic backbone for any computer vision model, replacing the convolutional layers and outperforming them with a large gap in terms of performance metrics such as top-1 accuracy and mean average precision (mAP) [93], and which has linear computational complexity with respect to the input image size.

Few-shot object detection has been a growing field lately but has not received as much attention as object detection for large-scale datasets or even few-shot image classification, mainly due to the task’s difficulty compared with image classification and the large variability of the models’ architecture in the object detection field [29]. Nonetheless, several proposals have been used in the field, such as meta-learning-based techniques [46], feature re-weighting [79] and fine-tuning-based approaches, such as the frustratingly simple few-shot object detection [167]

method, which is very similar to our method but which lacks its flexibility and applicability to other architectures.

1.2.4 Image Segmentation

Segmentation divides an image into distinct regions belonging to different categories (or objects). Segmentation is a rudimentary task in computer vision and has been a significant research field for the past few years. Traditional segmentation techniques include thresholding (Global or Adaptive), region growing, and edge detection. However, these techniques are limited in handling complex images where multiple objects overlap or are partially occluded, especially when the background is less controlled (real-life situations or drawings). Thanks to the big boom in deep learning after the successful training of CNNs in 2012, segmentation models based on convolutional neural networks (CNNs) have become the SOTA for segmentation (both for speed and accuracy), which have shown superior performance compared to traditional techniques. In cultural heritage preservation, segmentation can be used to detect and separate different objects in images, such as statues, buildings, and artifacts, which can help in preserving and documenting cultural heritage sites. There are two major sub-categories of segmentation which are Semantic Segmentation and Instance Segmentation.

Semantic

Semantic Segmentation is the simplest form of segmentation in which the goal is to attribute a class/category to each pixel. One of the most popular CNN-based semantic segmentation models is the U-Net, which was introduced in 2015 by Ronneberger et al [146] for the purpose of segmenting medical images. U-Net is an encoder-decoder architecture that consists of a contracting path that extracts features from the input image and a symmetric expanding path that generates the segmentation mask. The main difference between a traditional Auto-Encoder and U-Net is that U-Net uses only convolutional layers (no fully connected layers) and U-Net has a set of skip connections from each group of layers in the encoder to the decoder, allowing the U-Net decoding process to be detail-aware and more precise. In cultural heritage preservation, semantic segmentation can be used to extract meaningful information from images, such as the presence of specific objects, styles, and materials.

Instance

Instance segmentation, on the other hand, is a much more difficult task in which we have to assign a label corresponding to a particular class and a particular instance

of that class. Instance segmentation is a technique that requires the model to have the ability to differentiate between different objects and distinguish between different instances of the same object. Instance segmentation has gained significant attention in recent years, and various object detection models have been extended to perform instance segmentation, such as a Mask-RCNN [64], which is just a Faster-RCNN + a decoder. So in summary we can think of instance segmentation as object detection plus semantic segmentation.

1.3 Generative Adversarial Networks

Generative adversarial networks have changed how we think about artificial intelligence applications to real-life problems. Their ability to generate images is so realistic and impossible to distinguish from real images, which opened up many doors for applications, especially in cultural heritage. Generative, adversarial networks are not the first type of neural network that can perform the generation of new unseen hyper-realistic images. There are many previous attempts at this task that were quite successful but took too much time or lacked in the creativity department. What made GANs [34] so special is the idea of putting two neural networks in a competitive game in which they both get better with time. This is why we will discuss gans as two separate parts: the generative ability and the most essential part, the adversarial learning part.

1.3.1 Generative models

Generative models are now everywhere on social media and are the center of attention due to their ease of use and applicability. Currently, state-of-the-art models can only generate images and text, and audios and videos are still not at the state where they can be confused with actual data. Formally the definition of a generative model is a model that can capture the joint probability $P(X,Y)$ or $P(X)$ only in the case of unsupervised data. Generative models aim to include the data distribution itself and can tell you how likely a given sample is. A very simple example is the case of text generation, in which they can give you the probability of the next word in a sentence. Although text generative models are currently the most researched field, they are not the center of our thesis. Image generative models are also extremely powerful. As you can imagine, the concept of an image joint distribution is highly complex and even with neural networks it is very difficult to model such a distribution. Hence multiple types of generative models for images exist as we can see in [11].

- Explicit Density Estimators: aim to define and solve for $P_{model}(X)$. There are two types of explicit density estimators (Tractable, Approximate). Firstly,

Tractable EDE are sequential generators, and the most commonly known architecture for tractable EDE are PixelRNN and PixelCNN which use the chain rule to decompose the likelihood of an image X into the product of (many) 1-d pixel distributions. Both PixelRNN and PixelCNN use a neural network to perform the task of image generation in a sequential manner (PixelRNN with RNNs and LSTMs while PixelCNN uses CNNs). The only difference is PixelCNN is much faster in training than PixelRNN since it is more parallelizable. Both these architectures are super slow on inference time since they generate the image sequentially pixel by pixel. Secondly, Approximate EDEs, are models which are models that aim at modeling the full distribution of the data directly but in an approximate manner. Variational Auto Encoders are the most famous approximate EDEs and they aim at using an encoder decoder setup where the latent space is not just a simple embedding but a distribution of potential embeddings that we can sample from. Diffusion models are also a type of approximate EDEs and they are currently the SOTA of generative models as they generate HD images that are impossible to distinguish from real images. They do this by training and a denoising auto encoder 100 times on the same image but with an incremental increase of gaussian noise at each step, leading to the creation of a Markov chain Monte Carlo.

- **Implicit Density Estimators:** Aim at learning a model that can sample from $P_{model}(x)$ without explicitly defining it. As we have seen earlier, explicitly defining the data distribution was quite hard in earlier days of machine learning, with the results of PixelRNN, PixelCNN, and VAEs being super bad and very low quality since we didn't have enough calculations to build a good model. The idea was to skip the explicit part of generating images and aim at building a convertor that maps random vectors into images. Generative Adversarial Networks did exactly that, by sampling from a Gaussian distribution and mapping it into a generated image using a Decoder (Generator) setup. The generator is trained in an adversarial manner using a discriminator to find the different patterns between authentic images and generated images.

1.3.2 Adversarial Learning

Adversarial learning is the idea at the core of every innovation in this thesis. It is the most basic building block for most of the domain adaptation methods in the state of the art. Ever since the introduction of GANs, researchers have been fascinated by the idea of multiple neural networks collaborating together to build better models. And soon after the generative hype faded, researchers focused much

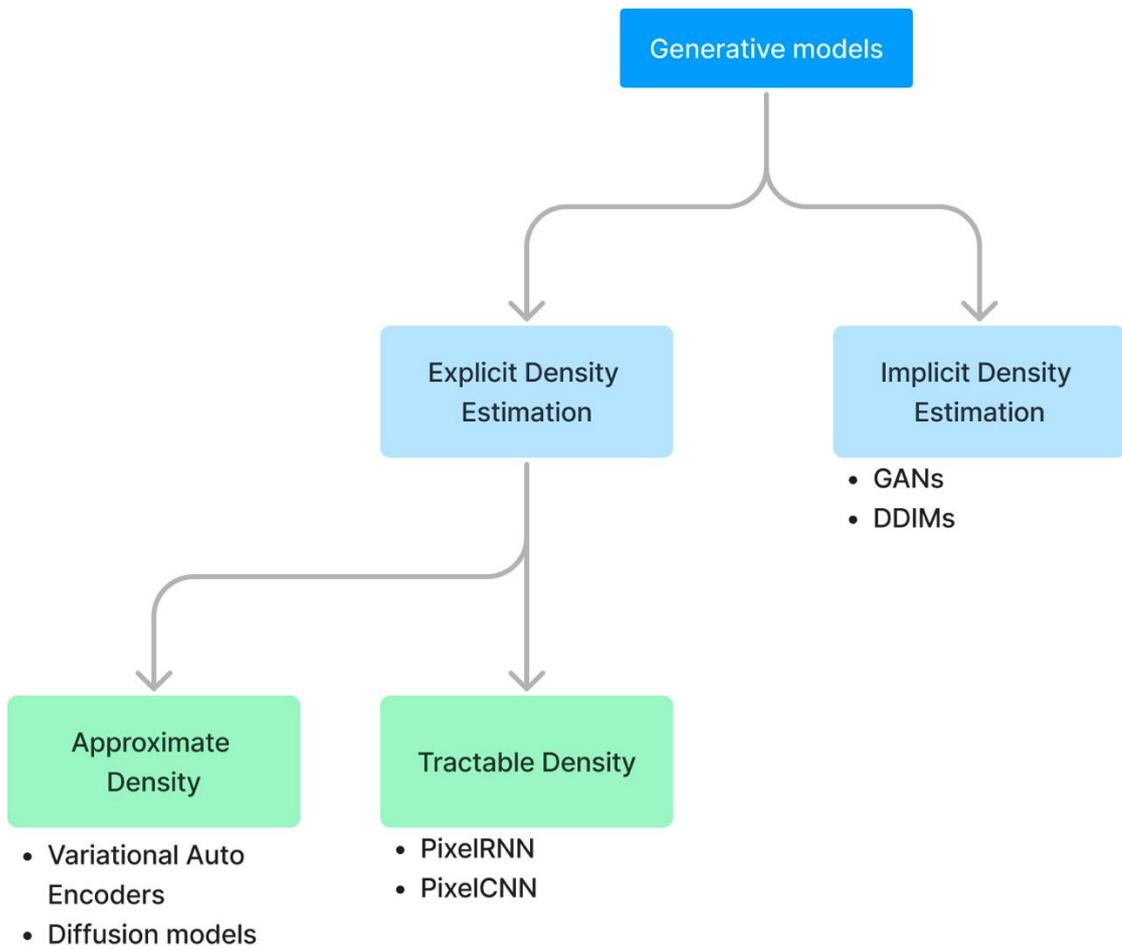


Figure 1.1: Cutting-edge generative model taxonomy.

more deeply on the adversarial side of the network as it unlocks a completely new way of thinking about building and training neural networks. The core idea is to define a behavior that is unwanted from your network (examples could be, a noisy latent space, a discriminative encoder, etc), build a component that aims to detect the presence of such behavior in your network, and then push the network in the opposite direction of this component. Similarly to when teaching a kid about the world, it needs a balance of positive reinforcement (this is an apple, this is a chair) but also it needs a teacher that tells it when it is wrong. Adversarial learning is usually done through the common cross-entropy, which was used for the GANs training loss, but it can be extended to any loss function with a little bit of creativity. Application areas of the technique are now endless from gaming and cyber security to robots and industrial machines.

1.3.3 Transfer Learning

Knowledge transfer or Transfer Learning is a large field of artificial intelligence that focuses on transferring knowledge from one domain (or task) to another. In most cases, the transferred knowledge is represented in the deep learning model's weights. Given the homogeneity of image classification models, which are based mainly on CNNs and Encoder/Classifier settings, Transfer Learning has advanced much faster than on object detection tasks with new research fields such as Domain Adaptation, Domain Generalization with several approaches: Instance-Based Approach, Feature-Based Approach, Parameter-Based Approach, and Relational-Based Approach. During the Transfer Learning process, some methods rely on various techniques: 1) unsupervised such as Image reconstruction, Adversarial losses [7], Image coloring, and Jigsaw puzzle solving [23]; or 2) Supervised such as Classification loss, Latent space-based losses, Pseudo-labeling and Separability Losses. The aim is usually to achieve two properties for the latent space of the input: (i) extract features from the data of both domains that a classifier can use to get good accuracy, i.e. Category Informative Latent Space; and (ii) make the latent spaces of both domains harder to tell apart, i.e. Domain Invariant Latent Space. Given the lack of object detection data, Y. Tang et al. [152] have tried to leverage image classifiers to build CNN-based object detectors. Such methods become harder to apply given the considerable heterogeneity of object detection architectures and their training procedures.

1.3.4 Domain Adaptation

Domain Adaptation has been one of the most active research areas in the last few years, and has been approached in both traditional Machine learning ways and more sophisticated Deep Learning based techniques. The deep Learning techniques

that were applied on DA varied a lot but they all aimed at achieving two properties for the latent space of the input: (i) extract features from the data of both domains that can be used by a classifier to get good accuracy i.e Category Informative Latent Space, and (ii) make the latent spaces of both domains harder to tell apart i.e Domain Invariant Latent Space. For this purpose many researchers have used Generative models to generate images from both domains aiming at finding a mapping between domains that allows the model to reduce the domain gap [68]. Only the discriminating portion of the Generate Adversarial Network has been used to formulate a minimization-maximization competition between the feature extractor (Encoder) and the domain discriminator that showed more promising results and faster convergence [11, 157].

Images are the primary source of information for computer vision models, which in essence aim to map an image into a category (or multiple categories). But images are quite hard to find in the field of MMS, unlike meta data which is very commonly found alongside every manuscript. This metadata can be represented in the form of a knowledge graph that can help restrain the computer vision model and allow it to better generalize to new data.

1.4 Metadata & Knowledge Graphs

Knowledge graphs (KGs) have emerged as a powerful tool for representing and reasoning complex knowledge in various domains in recent years. A knowledge graph is a graph-based knowledge representation that encodes entities as nodes and relationships between them as edges. We use Knowledge graphs in this thesis because they provide a rich source of structured knowledge allowing neural networks to incorporate information about the data allowing us to build more generic and controllable models. Knowledge graphs are a powerful tool that represents and reasons about the rich metadata associated with cultural artifacts in museums. IIIF provides a standardized format for representing metadata about digital images, which we transform into knowledge graphs to help neural networks perform their objective task more efficiently.

The International Image Interoperability Framework (IIIF) ⁵ is a set of open standards enabling digital image repository interoperability. IIIF is widely used in the cultural heritage domain, particularly in museums and libraries, to access their digital collections. We used IIIF as a tool providing a rich set of functionalities for interacting with high-resolution images of cultural artifacts, such as reducing image quality to speed up our calculations or even zooming at a region of interest. But the most important feature of IIIF is that it allows us to access meta-data

⁵<https://iiif.io/>

about the museum images in a standardized format, allowing for a far wider data collection. For example, museums can use IIIF formats to keep metadata about their manuscripts, including information about the creator, date, provenance, and description. This metadata can be represented as triples in a knowledge graph, with the manuscript as the subject, the metadata attribute as the predicate, and the metadata value as the object. Knowledge graphs can be used to perform various tasks, such as artifact classification, leveraging the rich metadata provided by IIIF.

1.4.1 Knowledge Graphs

In this sub-section, we go through a formal introduction to knowledge graphs, their usage, and how to create them. A knowledge graph is a graph-based knowledge representation tool (similar to a social media network) consisting of a list of entities, a set of relationships, and facts that connect entities through relationships.

An entity in a KG represents a real-world object or concept. For example, in our field of interest, the cultural heritage domain, an entity can be anything ranging from an artifact, a monument, a building, or a tradition to the central part of our thesis, which is a manuscript. A relationship in a KG represents a semantic connection between two entities. For example, in a cultural heritage domain, a relationship can be a temporal relation between two artifacts, a stylistic relation between two artworks, or a historical relationship between two monuments. Finally, a fact in a KG represents a triple that connects two entities through a relationship. For example, a fact can be (Mona Lisa, painted by, Leonardo da Vinci) in a cultural heritage domain.

KGs can be constructed from various sources, such as structured databases, unstructured text, and crowdsourcing. Constructing a KG involves entity extraction, relationship extraction, and fact extraction. Entity extraction involves identifying and extracting entities from the source data. Relationship extraction involves identifying and extracting relationships between entities. Finally, fact extraction involves constructing triples that connect two entities through a relationship.

MusicKG

MusicKG [43] is an example of the use of knowledge graphs in the field of cultural heritage and especially musical heritage. MusicKG is a multilingual KG specializing in medieval artworks relating to musicology and musical representations. MusicKG is the extract of multiple data sources such as Musiconis (collection of artworks from other museums), Musicastallis ⁶, Metropolitan Museum (NY)⁷,

⁶<https://musicastallis.huma-num.fr/>

⁷<https://www.metmuseum.org/>

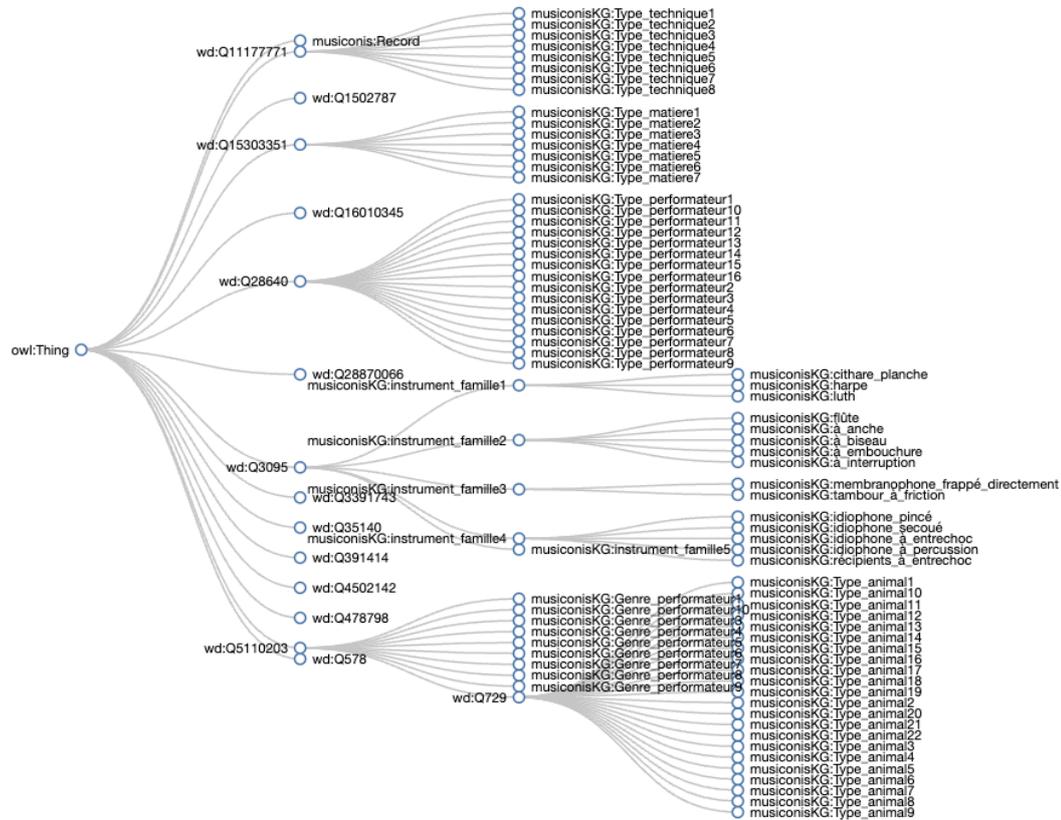


Figure 1.2: The hierarchy of concepts (entities) in the MusicKG knowledge Graph.

Mandragore⁸, Initiale⁹, Gothic Ivories¹⁰, and others. Each of these databases is specialized in a particular field of medieval music. Nevertheless, MusicKG built a shared knowledge graph for all of them, containing different concepts (hierarchies of concepts) and relations between them. Figure 1.2 shows the hierarchy of concepts in the MusicKG knowledge graph. Most concepts are similar to the W3C recommendations about the "Data on the web best practices."

1.4.2 Graph Embeddings & creation

Knowledge graph embeddings methods aim at mapping a component of a KG, including nodes, entities, and relationships between them, into a continuous vector space using algorithms such as Node2Vec. Machine learning models, mainly neural networks, can use the resulting vectors since this structure is simpler than a graph

⁸<https://mandragore.bnf.fr/html/accueil.html>

⁹<https://initiale.irht.cnrs.fr/>

¹⁰<http://www.gothicivories.courtauld.ac.uk/>

structure while preserving the inherent information and structure of the graph [165]. These techniques have gained popularity due to their wide utility for downstream applications such as KG completion and relation extraction, Recommender Systems, Question Answering, and Relation Extraction from texts.

1.4.3 Embeddings

Embeddings are an essential tool in deep learning, natural language processing, and, especially for us, in the field of Computer Vision. They allow for a sparse representation of high-dimensional data, such as words or images, in a rich and dense lower-dimensional space, typically intending to capture some underlying structure, meaning, or relations. Word embeddings, in particular, are a non-negotiable part of any natural language model, especially the deep learning models, as they are unable to input actual string words into a mathematical model. Hence the conversion from words and sentences into an embedding, which is a 1-D vector of float values that are more compatible with the type of operations performed in a neural network, and yet they capture the meaning and hopefully the context of the word too. One of the simplest and yet well performing methods for generating word embeddings is probably Word2Vec, which uses an additional neural network trained to learn words context by inputting a word and predicting its surrounding words, leading to develop latent spaces that are very rich in meaning and context information. These embeddings can be used for various tasks, from language modeling and machine translation to sentiment analysis and question answering.

Knowledge graph embeddings (KGEs) are considered to be a low-dimensional representations of the nodes and relations in a knowledge graph. Knowledge graph embeddings are mappings on different parts of the knowledge graph into a vector space that satisfy certain properties and maintain the information that exists in the graph. Each method defines a score function which measures the distance of two nodes relative to their relation in the mapped embedding space. These goal of these score functions can be summarised as keeping the nodes which are connected to each other in the graph close in the mapped dimension and those which are not connected far from each other. The most famous score functions are TransE, TransR, RESCAL, DistMult, ComplEx, and RotatE [165].

Knowledge Graph embeddings aim to embed components of a KG, including entities and relations, into a one-dimensional continuous vector space that can be used to train many types of models, such as link prediction, triple classification, entity classification, and more. The majority of the knowledge graph embedding techniques fall into two categories:

- Translational Distance Models: which exploit the distance-based vector scoring functions. Such as TransE [165], which was previously mentioned that

represents both entities and relations as one-dimensional vectors in the same space. Taking a fact (h, r, t) , the relation is represented as a translation vector r so that h to t can be as similar to r as possible (with low error), similar to `word2vec`.

- Semantic Matching Models: these exploit similarity-based functions that measure the plausibility of facts by matching latent space semantics of the tuple's components represented in their vector spaces, Such as RESCAL [115], [165], which represents each entity with a vector and each relation with a matrix representing the pairwise interactions between the factors, which are later decomposed using rank- k decomposition through solving an optimization problem that is correlated to minimizing the error of the decomposition reconstruction.

Now that we have discussed the different types of knowledge graph embedding techniques we need to clarify the two main assumptions that are often made regarding data availability.

- The closed world assumption (CWA) [85]: assumes that anything that is not pre-known is false, meaning that it is sufficient to detect the absence of information to determine that it is false. This is commonly used in situations where we assume that the data is complete and all needed information is available, as in the medical field where we assume that all patients do not have any disease that is not mentioned.
- the open world assumption (OWA) [180]: doesn't make any assumptions on the veracity of a fact if it is not mentioned in the data. Meaning the absence of information is not evidence that the information is false. Most of the knowledge graphs in the field of culturale heritage are missing vast parts of the data from missing meta-data, lack of attention or the manuscript it self is destroyed hence we can't recover all the answers.

1.5 Conclusion

This chapter represents a summary of most of the works that either influenced this thesis or are related to it. Firstly, we described the advancements in the field of Medieval Studies and cultural heritage and the major lack of recent AI advancements in the field. Secondly, we described the computer vision field in detail, both at the level of state-of-the-art applications and research but also defining some very clear concepts that we will be referencing throughout the thesis, such as object detection models and the evaluation methods for object detection. Thirdly, we discussed the generative adversarial networks and finished with a few definitions for Knowledge Graph Embeddings.

Chapter 2

Datasets

This chapter focuses on the first problem of the thesis, which is missing data. The datasets chapter is a very important part of the thesis, as it will be referred to in every other chapter throughout the thesis. We will describe all the datasets that we annotated and filtered through things like classes and distributions.

2.1 Introduction

All around the world, there are people with different cultures, religions, and languages. However, they have one point in common: music as a means of expression and communication. Music is a universal language used since ancient times by human beings to express all kinds of emotions and feelings. By analyzing the evolution of musical instruments throughout history, we can understand, share, and value the musical heritage left by our ancestors.

Several public and private collections (such as the Music Museum¹ of the Paris Philharmonie in France, the Gallery of the Academy of Florence² in Italy, or the J. Paul Getty Museum³ in USA) preserve musical instruments from Modern history (from 1500 to the present). However, only a few musical instruments from the Late Antiquity (4th to 6th centuries AD) or the Middle Ages (5th to 15th centuries AD) are still conserved nowadays. The older the musical instrument, the fewer well-preserved copies of that instrument are found. Those old musical instruments that have not been preserved can only be studied thanks to their representations in artworks and illuminated manuscripts.

To preserve cultural heritage, different governments, and public institutions have carried out massive conservation programs. These programs not only seek to

¹<https://philharmoniedeparis.fr/en/musee-de-la-musique>

²<https://www.galleriaaccademiafirenze.it/en/>

³<http://www.getty.edu/museum/>



Figure 2.1: An image segmentation example showing a female musician playing the vielle, female and male dancers from the AnnMusiconis database

create the optimal conditions to preserve the manuscripts in their physical form but also digitize them to disseminate their content. The IIF is a clear example of a joint international initiative to share images freely. It is estimated that more than one billion IIF images are available nowadays, and about 400 million of them date from the Middle Ages or before. Even though online search engines enable browsing these collections, only the bibliographic data of the manuscripts is indexed (such as title, date, author, and origin), and the page’s content is not considered. Thus, thousands of images should be manually scanned by experts until a new instrument representation is found, making this search an arduous and time-consuming task.

Hence, our first major contribution in the thesis is to find and annotate multiple datasets in the field of cultural heritage and especially medieval music. In this chapter, we will present AMIMO, AnnMusiconis, AnnVihuelas, Medieval Musicological Studies Dataset, which are all datasets that we manually annotated throughout the thesis. To this end, we first review related works on illuminated and historical image datasets. And then, we present each major dataset we annotated in detail.

2.2 State of the art

Publicly available datasets allow the evaluation and comparison of different neural algorithms. Several historical manuscript image datasets [120, 147, 143] have been proposed with the goal of training and evaluating neural networks. This section presents an overview of existing illuminated and historical image datasets.

2.2.1 Illuminated manuscripts datasets

Ancient musical instruments (e.g., vielles, lutes) can be found in illuminated manuscripts. Among the existing illuminated manuscripts datasets, we can mention the HBA corpus [109], which is a pixel-based annotated dataset created for the ICDAR'2017 Competition on Historical Book Analysis. The HBA dataset contains 4.436 real scanned ground-truthed historical document images from 11 books in different languages and scripts published between the 13th and 19th centuries. Another historical dataset is HORAE [19], a dataset of annotated pages from books of hours (a type of handwritten prayer books owned and used by rich lay people in the late middle ages).

2.2.2 Historical manuscripts datasets

A large number of annotated datasets of historical manuscripts have been created for document analysis [5, 141]. However, most of them are not illuminated, and only the text is annotated. For example, the MLM (Multiple Languages and Modalities) dataset [4] is a resource to train and evaluate multitask systems on samples in multiple modalities and three languages. In [90], the authors present the Newspaper Navigator dataset containing 16.3 million pages from digitized historical newspapers in the USA. The BADAM dataset [83] is a corpus of 400 annotated page scans of Arabic and Persian manuscripts spanning a wide range of topics and dates of production. The Pinkas dataset [6] introduces a public historical document image dataset. It is the first dataset in medieval handwritten Hebrew and fully labeled at word, line, and page levels by experts on historical Hebrew manuscripts.

Although the document analysis community has made an enormous effort in image liberalization and bounding box creation, there is a lack of specialized ground truth to validate computer vision methods for illuminated manuscripts. Moreover, the datasets containing historical document images have some particularities [109], such as the superimposition of information layers (e.g., signatures, stamps, handwritten notes) and the variability of their contents and noise (e.g., different layout, typography, font styles, scanning) that difficult their collection and annotation process.

In summary, even if several image datasets exist for image segmentation, none has become a standard benchmark since they lack diversity and completeness. According to Kiesel et al. [82] the issues that prevent the reuse of the existing datasets include missing data sources, bias due to heuristic annotations, no ground truth annotations, unavailability, and a non-representative sample. The creation of annotated illuminated manuscript datasets remains an open issue, and new computer vision methods and ground-truth data are needed to tackle image segmentation problems.



Figure 2.2: Examples of chordophones in the MIMO database.

2.3 Annotated MIMO (AMIMO)

An annotated image dataset of musical instruments for organological analysis and image segmentation. This dataset contains 10.258 manually-created annotations of the existing chordophones' images in the MIMO dataset. MIMO ⁴ stands for "Musical Instrument Museums Online" and is the largest freely accessible image database of musical instrument collections held in public collections worldwide. The images are accessible at <https://mimo-international.com/MIMO/> and our annotations are placed in the public domain for unrestricted reuse. The AMIMO dataset will be available for download at <https://github.com/ImadEddineBek/AMIMO> with a link to a ROBOFLOW dataset that allows immediate training and getting started with the results of the thesis for tasks such as: Object detection, Image segmentation. We show samples of the MIMO dataset [2.7](#), the first image shows a zither ⁵, the second one shows a qanun ⁶, the third shows an Harp ⁷ and the last one ⁸ shows a Violoncello.

2.3.1 The MIMO dataset

MIMO stands for Musical Instrument Museums Online. The MIMO database <https://mimo-international.com/MIMO/> is the world's largest freely accessible database for information on musical instruments held in public collections. Initially, the MIMO consortium started as an initiative of the most important musical instru-

⁴<https://mimo-international.com/MIMO/>

⁵https://mimo-international.com/MIMO/doc/IFD/OAI_SMS_MM_POST_6953

⁶https://mimo-international.com/MIMO/doc/IFD/OAI_MDMB_309797

⁷https://mimo-international.com/MIMO/doc/IFD/OAI_CIMU_ALOES_0157264

⁸https://mimo-international.com/MIMO/doc/IFD/MINIM_UK_5000



Figure 2.3: Zither-Harp - Made in Germany - 1942



Figure 2.4: Qanun - Türkiye - 1825



Figure 2.5: Harp 40 strings.
7 pedals. Hook mechanics.
Paris 1783.



Figure 2.6: Violoncello - London - 1792

Figure 2.7: 4 examples of images from the AMIMO dataset.

ments museums in Europe. The objective was to create a single online access point to their collections. This initiative was founded by a European Commission project between 2009 and 2011. The idea was not only to share their digitalized musical instrument collections but also to create common terminology and a unified classification system. In addition, the consortium created standards for photographing musical instruments and detailed guidelines to digitize their collections. Figure 2.2 depicts five images of string instruments in the MIMO dataset:

- **Harp** : the first example is an harp⁹ from the beginning of the 19th century currently in Paris, France.
- **Lute**: the second example is a lute¹⁰ from 1721 belonging to a museum in Prague, Czech Republic.
- **Lyre/guitar**: A lyre-guitar example¹¹ from 1809 conserved at the Musee de la musique in Paris, France.
- **Rebec**: The image of a rebec¹² from 1905 held in Cologne, Germany.
- **Zither-harp**: Finally, a zither-harp example¹³ from 1902 nowadays preserved in Vienna, Austria.

Due to the tremendous success of this initiative, MIMO now contains collections from all around the world and has become the reference resource for musical instruments. Nowadays, MIMO relates the most prestigious instrument museums worldwide, such as the Philharmonie de Paris, the University of Edinburg, the MIM de Bruxelles, the Galleria dell'Academia, or the Museu de la Música de Barcelona. MIMO harvests the digitalized images jointly with detailed musical instrument descriptions available in six languages. The access is free and multilingual.

2.3.2 String instruments

Musicologists rely on the organological classification called Sachs-Hornbostel, which was updated by MIMO, to define the following instrumental families according to the sound made by the instrument:

- **Idiophones** : the sound comes directly from the instrument's material (example: bells).

⁹https://mimo-international.com/MIMO/doc/IFD/OAI_CIMU_ALOES_0157264

¹⁰https://mimo-international.com/MIMO/doc/IFD/OAI_ULEI_M0002908

¹¹https://mimo-international.com/MIMO/doc/IFD/OAI_CIMU_ALOES_0130404

¹²https://mimo-international.com/MIMO/doc/IFD/OAI_ULEI_M0002704

¹³https://mimo-international.com/MIMO/doc/IFD/OAI_ULEI_M0003518

- **Membranophones** : the sound is produced by the vibration of a stretched membrane (example: drum).
- **Chordophones** : the sound is produced by the vibration of one or more strings, stretched between two points (example: harp).
- **Aerophones**: the sound is created by a vibration due to the passage of air.

These categories make it possible to avoid naming the instruments more precisely because the names of the instruments often change from one region to another, from one language to another, especially for the medieval period, where there is no unified lexicon for musical instruments.

The identification of ancient chordophones from two-dimensional images is a challenging task. Therefore, experts in medieval musical iconography are often required to verify the instrument’s annotations. Among all the musical instrument images in the MIMO database, we decided to annotate the chordophones family in this study.

In particular, string instruments plucked by the fingernail or by a plectrum, or rubbed with a bow. The presence of the bow in the musical scene is essential to distinguish the bowed strings since some forms of instruments are quite similar. To better characterize ancient instruments, we subdivided the chordophones family into four types: lute, vielle, harp, zither.

- **Lute type** : (number 321 in the Sachs-Hornbostel classification) describes instruments with a neck as an extension of the soundbox (cut in the mass or nested). The strings are stretched from the soundbox to the pegs at the end of the neck. The strings are plucked (**lute type**) or bowed (**vielle type**).
- **Harp type**: (322 in Sachs-Hornbostel) designates instruments whose strings are stretched perpendicular to the soundbox. The strings are tuned by pegs located on the console, which is generally in the upper part of the wooden triangle with a vertical column and the soundbox placed obliquely.
- **Zither type** : (314.1 in Sachs-Hornbostel) refers to instruments whose strings are stretched parallel to the soundbox, either totally like the psaltery (**psaltery type**) or partially like the crwth or certain lyres (**lyre type**).

2.3.3 Annotation Process

The AMIMO dataset (Annotated Musical Instrument Museums Online) is a high-quality annotated image collection of historical musical instruments. This dataset provides an excellent opportunity for deep learning to understand in-depth the

characteristics of these instruments. The dataset aims to provide an annotated dataset for pre-training a large model for historical musical instrument recognition for different tasks, such as object detection, instrument recognition, and mainly instrument segmentation.

The dataset contains information about the objects: the name of the instrument, origin, location, date of creation, title of the host page, URL of the source description, and the image URL (Full size/w-250). Five experts in organology annotated the dataset following three steps: Object contouring, Object classification, and classification verification.

Object contouring in AMIMO

The first step of object segmentation was built using the supervisely¹⁴ (a tool for image annotation that allows online collaboration for large teams) polygon contouring technique. This method allows the user to highlight an object in the image by defining its borders using a set of points. Thus, the user creates a concave hull surrounding the object as precisely as possible. Since the annotated objects do not have an empty part in their centers, the borders can correctly define them.

Our experts manually annotated the polygons. Those polygons are converted later into either a bounding box or an object mask to train different models.

Object Classification

The second step is the Object classification. After the objects are selected, our musicologists separately annotate each object and classify it in one of our six categories. This step allows us to detect the objects that are easy to classify and the more difficult ones.

Classification Verification

The last step is to verify the classes. In this step, all the experts are gathered together to discuss each complex object in detail. Then, based on their expertise and the available information, they achieve a consensus to annotate those complex objects.

2.3.4 Class distribution

Our dataset contains an extensive list of images covering from 1600 to the late 2000s. The main characteristics that make our dataset useful for pre-training larger neural networks that will operate on smaller musical datasets are the following:

¹⁴<https://supervise.ly/>



Figure 2.8: An image segmentation example showing an annotated French Harp from the 1790's in the MIMO dataset.

Table 2.1: The distribution of objects in AnnMusiconis across different classes.

Class	Counts
Vielle	3508
Lute	3163
Zither	2102
Harp	867
Bow	437
Lyre	181
TOTAL	10.258

1. Diverse instrument types: musical instruments evolve through history. Due to the large historical period covered by the data, we work with a huge collection of different instruments for the same class and also different classes (as shown in table [2.1](#)).
2. Large object to image ratios: most of our instruments cover a large part of the image since they are the focus of digitalization. Therefore, these images are perfect for learning the structure and form of the instruments and generalizing them to other datasets.
3. High-quality images: the images we work with are of high quality, as big as 3000*2000 pixels.
4. Large geographical coverage: The data we are using covers 11 different countries.
5. Knowledge Graph-like structure: We associate to our dataset a set of useful information about the context and the background of the instruments, such as their original location and creation date.

2.4 AnnVihuelas

The Vihuelas dataset ¹⁵ is the only dataset that we didn't create manually and was collected by John Griffiths. The dataset is specialized in the vihuela, a stringed instrument that was popular in Spain and Portugal during the Renaissance. We annotated 165 chordophones in total. Since it is a dataset of vihuelas, most of the annotated instruments are from the lutes family.

We show a sample of the vihuelas database in [2.18](#), the first picture ¹⁶ shows an angel playing waisted sides, long neck, 4 strings, curved pegbox terminating in an animal head vihuela. The second shows a sculpture made of baked clay and polychrome, painted in ochre in 1792 and restored in 1912, this vihuela might represent the "missing link" between the medieval guitarra (3 course of double or triple strings) and the 4-course guitar of the 16th century. The third ¹⁷ shows an excellent example of a vihuela with cornered waists, in Italian, a viola da mano. The instrument has 6 courses of strings matched by 12 pegs on its sickle-shaped pegbox, one of the main features that distinguishes it from similar Spanish depictions of the vihuela c. The fourth shows an angel with 5-course guitar. the instrument is similar to many 16th-century vihuelas, although the decoration around the sound hole is characteristic of guitars of the first half of the 17th century.

¹⁵<https://vihuelagriffiths.com/>

¹⁶<https://vihuelagriffiths.com/vihuela/instruments/25477/>

¹⁷<https://vihuelagriffiths.com/vihuela/instruments/25483/>



Figure 2.9: Angel playing vihuela
 - 13cent - vihuela de péñola
 - Salamanca, Catedral vieja,
 Capilla del Aceite



Figure 2.10: Angel musician with three-
 string vihuela or guitar. Lorenzo Mer-
 cadante de Bretaña, 1464-1467



Figure 2.11: Viola da mano played
 by Serafino Aquilano



Figure 2.12: Angel guitarist from “La
 Presentación de Jesús en el Templo.”
 painted by Diego Valentin Díaz. 1600-
 1650.

Figure 2.13: 4 examples of images from the Vihuela dataset.

2.5 AnnMusiconis

The Musiconis database ¹⁸: was built by historians and musicology experts to analyze musical performances featuring instrumental musicians (Wind Instruments, Percussion Instruments, Stringed instruments, etc), singers, and dancers present on medieval objects (Manuscripts mainly, but also Stained glass, Stone sculpture, Ivory sculpture, Wood sculpture, or Engraving) from the 8th to the 16th century. We annotated a smaller segment of 662 chordophones to stay consistent with the AnnVihuelas dataset, a small Renaissance chordophone. The image distribution is as follows: 112 are Citharas, 132 harps, 56 lutes, 75 lyres, and 327 vielles (217 vielles played with a bow).

We show samples of the AnnMusiconis dataset [2.7](#). The first image shows a vièle ¹⁹, the second one shows a vièle painting ²⁰, the third shows a luth ²¹ and the last one shows king David playing the Harp ²².

Musiconis was developed with financing from the French National Research Agency (ANR) by a team of researchers from the University of Paris-Sorbonne, the University of Poitiers, and the CNRS. The database is freely accessible online to researchers, students, and the general public, and is continuously being improved by musicology master’s students, and experts. Our hope is that with the help of our AI, Musiconis will benefit greatly from new resources.

The creation of Musiconis was motivated by the need to provide a comprehensive resource for studying musical performances in the Middle Ages. Prior to the creation of Musiconis, there was no single database that brought together images of musical performances from a variety of sources. This made it difficult for researchers to study musical performance’s evolution over time and compare different cultures.

The creation of Musiconis was a significant undertaking. The researchers had to identify, collect, and digitize images from various sources, including manuscripts, sculptures, paintings, and stained glass windows. They also had to develop a system for annotating the images with metadata, such as the date, location, and type of performance.

¹⁸<http://musiconis.huma-num.fr/fr/>

¹⁹<https://musiconis.huma-num.fr/fr/fiche/47/roi-david-jouant-de-la-viele-en-huit.html>

html

²⁰<https://musiconis.huma-num.fr/fr/fiche/1920/anges-jouant-de-la-viele-du-luth-de-la-chalem.html>

html

²¹<https://musiconis.huma-num.fr/fr/fiche/529/putto-jouant-du-luth.html>

²²<https://musiconis.huma-num.fr/fr/fiche/442/initiale-historiee-ps-1-representant-le-roi-da.html>

html



Figure 2.14: King David playing the vièlegin with Saints Michael and Benedict - 1151 - Troyes, Grand Est, France.



Figure 2.15: The Assumption of the Vir-
1493 - Cortona, Toscane, Italie



Figure 2.16: Putto Playing the luth -
1540 - Auch, Occitanie, France



Figure 2.17: King David playing the harp
-inspired by Saint-Esprit - 1200 - Paris,
France.

Figure 2.18: 4 examples of images from the Muiconis dataset.

2.5.1 Types of Images

Musiconis contains over 2800+ images of musical performances from the Middle Ages. The images come from a variety of sources, including:

1. Manuscripts: Musiconis contains images of musical notation from manuscripts, which can be used to study the development of musical notation over time.
2. Sculptures: Musiconis contains images of sculptures depicting musical performances, which can be used to study the social and cultural context of music in the Middle Ages.
3. Paintings: Musiconis contains images of paintings depicting musical performances, which can be used to study the artistic representations of music in the Middle Ages.
4. Stained glass windows: Musiconis contains images of stained glass windows depicting musical performances, which can be used to study the religious and liturgical aspects of music in the Middle Ages.

2.5.2 Meta Data

The images in Musiconis are annotated with a variety of metadata, including:

1. The date of the image
2. The location of the image
3. The type of performance depicted in the image
4. The instruments used in the performance
5. The singers and dancers depicted in the performance
6. The relationships between the performers
7. The religious or liturgical context of the performance
8. Title in English, French, and Spanish
9. Instrument specific information (Material, Stem type, Horn type, mouthpiece type, presence of holes, number of strings, etc).

This metadata allows researchers to study the images in Musiconis in a variety of ways. For example, researchers can use the metadata to track the evolution of musical instruments over time, to compare different cultures, or to study the role of music in religious and liturgical contexts.

We include an image from Musiconis in [2.19](#)²³.

²³<https://musiconis.huma-num.fr/en/fiche/1295/shepherd-holding-a-bagpipe.html>

Shepherd holding a bagpipe



Summary

English title : Shepherd holding a bagpipe

Location (current) : Tours, Centre-Val de Loire, France

Location type : Library

Location (original) : France, , France

Century : 14

Dates : 1320 -1330

Technique : Illumination

Material : Paper / Parchment

Partner database : Initiale

Original title : Annonce à un berger

Links :

<http://initiale.irht.cnrs.fr/decors/decors.php?id=8655>

Figure 2.19: A screenshot from Musiconis website of a Shepherd holding a bagpipe.

2.5.3 Classes

AnnMusiconis contains a large list of instrument types, Idiophones, Membranophones, Chordones and Aerophones. For the sake of simplifying the scope of the thesis we decided to work only on String instruments (Chordophones). We showed the list of objects we work with in table [2.2](#).

accessed on 15th of August 2023.

Table 2.2: Distribution of objects in our AnnMusiconis across different classes.

Class	Counts
Citharas	112
harps	132
lutes	56
lyres	75
vielles	327
TOTAL	662

2.6 Medieval Musicological Studies Dataset (MMSD)

Throughout the thesis, I collaborated closely with Valérie LEPAGE (musicologist with IReMus) and she visualized hundreds of IIF manuscripts from various collections to find images of illuminations with singing performances. Among the consulted institutions were the French National Library (BnF), the J. Paul Getty Museum, the Universitätsbibliothek Basel, the University of Cambridge, and the University of Princeton. As a result, we obtained a dataset of 341 IIF images of illuminations. In total we annotated, 693 objects, with Book 338, Phylactery 204, Lectern 87, Altar 37, Folio 27.

The image dataset creation and the ground truth annotation and validation processes were organized into four different steps, described as follows.

1. image dataset: The first step was to create a dataset of images for training. Therefore, three experts in musicology and professional singers searched for and manually selected images of manuscript pages containing singing representations.
2. Annotate objects: As the domain expert, Valerie LEPAGE annotated the dataset using the Supervisely tool (<https://supervise.ly/>, accessed on 1 November 2021), which is a collaborative online tool for image annotation, allowing users to create bounding boxes and object masks. As a result, the objects (such as books, lecterns, altars) in each image are highlighted by defining its borders.
3. Classify the objects: In the third step, the objects annotated previously were manually classified as book, folio, phylactery, lectern, or altar by the musicologists. Thus, we can not only detect objects but also the exact position of those objects within the image.
4. Obtain a consensus in the classification of objects: As we explained previously, it is not easy to detect singing performances. We are working with images of artworks, so the singing representations are not real; they are paintings or drawings of an artist who does not necessarily know about vocal practices. Therefore, the fourth and last step in the classification of objects consists of achieving a consensus among all the experts to create the ground truth.

2.6.1 Image Dataset of Illuminations Representing Medieval Singing

At the beginning of the Middle Ages, musical instruments were mainly used as a complement for singing. At that time, in religious music, the vocal song was



Figure 2.20: An annotated phylactery example

thought to represent the divine Word of God. Therefore, religious music was promoted and developed by high authorities of the Church. Medieval singing was present in all services, major festivals, and ceremonies. Religious songs were repeated every day in churches and monasteries. In secular music, singing performances were also of vital importance. Secular vocal songs were generally transmitted orally. They reflected ordinary people's daily lives, love and war stories, or songs intended for processions going to battle.

The importance of the notation of songs is also manifested by rich illuminations, which gradually range from simple colored to intricately decorated capitals and complex scenes with multiple characters and rich details. These richly illuminated manuscripts provide musicologists with numerous clues concerning the practice of singing. Analyzing a large dataset of images may reveal previously unknown details to clarify medieval singing beyond eras and regions.

Massive amounts of digital iconographic data are available nowadays thanks to the development of the IIIF standard. The IIIF standard offers unified access to view and read digitized ancient documents, significantly increasing the number of people who can consult them. The most important cultural institutions all around the world publish their collections using the IIIF format. The vast digitization of manuscripts now makes it possible to reconsider iconographic studies and analyze a series of gestures and behaviors that can be compared with narratives,

Table 2.3: Distribution of objects in MMSD across different classes.

Class	Counts
Book	338
Phylactery	204
Lectern	87
Altar	37
Folio	27
TOTAL	693

descriptions, and texts of treatises. Thus, new illuminated medieval collections are now available for researchers, allowing them to harvest millions of illuminated manuscripts. The ever-increasing number of singing performances hidden in those illuminated manuscripts challenges researchers to develop new pattern-recognition methods to find them.

2.6.2 Annotation of Written Supports in Illuminations

The presence of books is an essential aspect of liturgical song representations. Although in the Middle Ages [72], songs were mostly memorized since they were repeated every day in ceremonies and mass, some songs required different texts depending on the days and times of the liturgical year. Therefore, due to the increase in the repertoire and the development of polyphony, it was necessary to create written supports (in this chapter we mention support as an object that contains writing in general whereas the rest of the thesis a support is the object of the image - manuscript, painting, glass, sculpture). Examples of written supports are missals and graduals. The size of written supports may vary from a hand-held book to a large codex requiring a lectern, around which the singers would gather, in the middle of the choir. Written supports could be found in the hands or knees, tables, altars, lecterns, palace hall, or gardens.

Another type of written support for singing performances is phylacteries and sheets (with or without musical notation). There are well-known phrases that are always sung, such as “Ave Maria Gratia plena”, “Cantate Domino”, or “Gloria in Excelsis Deo”. When these phrases are found in phylacteries or the texts around illuminations, they indicate a singing performance.

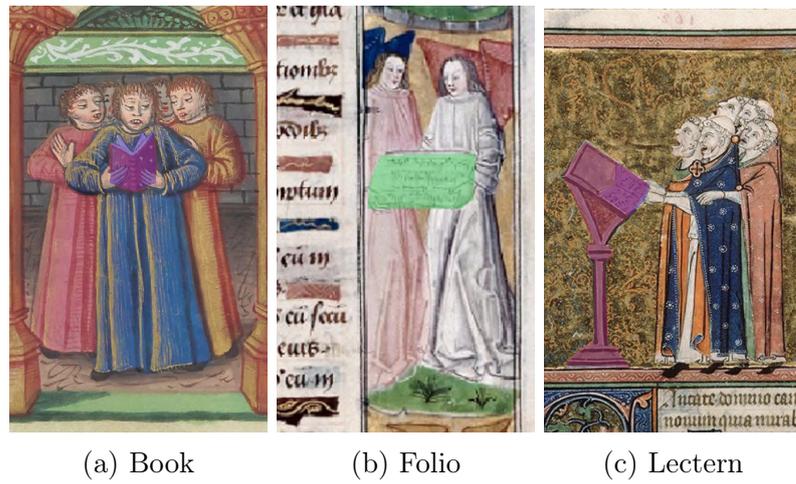
Therefore, in a previously selected dataset of IIF images selected manually by experts containing singing performances, we annotated the following classes as described in Table 2.3:

1. Phylactery:: a medieval speech scroll, which contains or depicts speech, song,

- or other sounds (for example, Figure 2.21d (Bodmer ms. 91 f. 39r: http://www.e-codices.unifr.ch/loris/fmb/fmb-cb-0091/fmb-cb-0091_039r.jp2/full/full/0/default/jpg, accessed on 1 November 2021)).
2. Folio: a thin and flat surface that can be used for writing or drawing. (Figure 2.21b (Abbeville ms. 016 f. 15: https://iiif.irht.cnrs.fr/iiif/France/Abbeville/B800016201/DEPOT/IRHT_106357_2/1000,500,800,1500/full/0/default.jpg, accessed on 1 November 2021)).
 3. Book: a collection of sheets bound together containing printed or written texts, pictures, etc. (Figure 2.21a (BnF ms. fr. 166 f. 119v: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b105325870/f252/1500,3100,1000,1200/full/0/native.jpg>, accessed on 1 November 2021)).
 4. Altar: a sacred table used for ritual sacrifice or offerings in a religious building. (Figure 2.21e (Initiale ms. 3028 f.082 <http://initiale.irht.cnrs.fr/decor/58000>, accessed on 1 November 2021)).
 5. Lectern: a reading desk with a slanted top, on which books are placed for reading aloud. (Figure 2.21c (Avignon ms. 0121, f. 162v: https://bvmm.irht.cnrs.fr/consult/consult.php?mode=ecran&reproductionId=15460&VUE_ID=1393519, accessed on 1 November 2021)).

2.7 Conclusions

As artificial intelligence reaches new limits, the most logical extensions are no longer model improvements but applications of such technologies in new fields and domains. Hence our first contribution in this thesis is the datasets manually annotated by Valerie LEPAGE and other musicology experts in the hopes of pushing research ahead. In this chapter, we presented four datasets, Annotated MIMO, a dataset of actual pictures of real instruments, AnnMusiconis, a dataset of manuscripts and sculptures of artistic medieval instruments, AnnVihuelas, a dataset for the vihuelas instrument, and finally, Medieval Musicological Studies Dataset (MMSD), a dataset of medieval signing. In the next chapters, we put these datasets into use.



(a) Book

(b) Folio

(c) Lectern



(d) Phylactery

(e) Altar

Figure 2.21: Examples of object annotations.

Chapter 3

Dual Training for Transfer Learning

The methods and some results presented in this chapter were published in:

I. E. I. Bekkouch, V. Eyharabide and F. Billiet, "Dual Training for Transfer Learning: Application on Medieval Studies," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: <https://doi.org/10.1109/IJCNN52387.2021.9534426> [14].

This chapter presents a new method for non-intrusive Transfer Learning that allows for fast improvement of performances over vanilla Transfer Learning while remaining simple, reproducible, model-independent, and task-independent. Although most non-intrusive TL methods tend to separate the two domains and their training, our method's strength lies in combining both sources in mutual training of the same model architecture.

3.1 Introduction

Understanding and preserving cultural heritage is essential for comprehending our past and influencing our present. With regard to everything from music and art to architecture, theology, and more, the Middle Ages represent a dynamic and diverse period in our history. It is challenging to go through all the information we have about that period because studying medieval manuscripts requires extensive experience and historical knowledge, leaving much to be discovered. Fortunately, current developments in artificial intelligence and computer vision offer a way to automate the procedure and quickly extract valuable information from enormous amounts of data.

As of 2015, deep learning-powered computer vision models beat human performance on the ImageNet Large-Scale Visual Recognition Challenge 2014 [132] picture classification challenge using models like ResNet [65] and GoogLeNet [151]. Following that, improvements persisted in the form of bounding box regression

models, which forecast both the class and the location of the object on the image. Due to the fact that these models were designed solely for single-object images, a new research area dealing with multi-object identification was created. These models' two most well-known branches are Detectron [172, 55] and YOLO [123, 121, 122, 18, 76]. Even on tiny datasets, these models produce reliable findings when applied to contemporary photos. However, when we apply them to pictures of antique sculptures and paintings, their performance typically suffers significantly. Such issues arise as a result of the sort of data that YOLO and other object identification techniques were trained on, resulting in what is known as a domain gap issue [11].

The most common types of domain gaps in research are photos of objects spanning different camera types [135], datasets collected with different composition biases [154], or different abstractions of the objects [91]. The datasets we use in the thesis present a challenge to machine learning models for four main reasons:

- 1. Variation in style:** Our dataset covers musical instruments from past and recent history. The instruments also come in different supporting materials such as paintings, manuscripts, photographs, and sculptures.
- 2. Difficulty in acquiring and labeling:** When analyzing ancient and damaged artworks, experts sometimes have difficulties recognizing objects in images. Building Object detection models for such historical data is a challenging task that requires an experts' consensus.
- 3. Scarcity of data:** Manually annotated images of medieval artworks are scarce, even more in the medieval musicology domain.
- 4. Small Regions of Interest:** As with many datasets of object detection, our dataset focuses on a small region and not the image as a whole, making the detection problem more challenging (as shown in Fig. 3.1¹).

Transfer Learning is the first answer to mind when facing such domain gap issues. It consists of using a pre-trained model on a source task and transferring the learned knowledge into a target task. Typically, the original model is trained on a large dataset with many classes, which allows it to learn discriminative features for a large number of tasks. The first part of the model is usually kept the same since it leans toward generic discriminative abilities. In contrast, the final layers either require updating or a full change and re-train depending on whether the target and the source tasks are the same. However, Transfer Learning in its vanilla form has its limitations and tend to work only on datasets with core style similarities.

¹<http://musiconis.huma-num.fr/fr/fiche/70/musiciens.html>

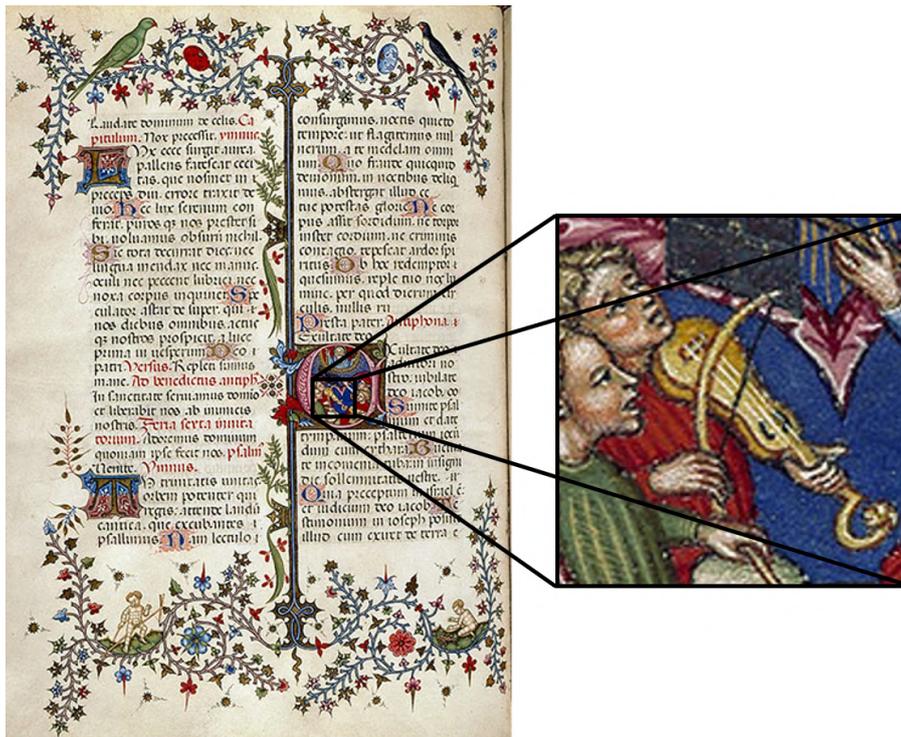


Figure 3.1: An example of a Musiconis' image to show the small size of the musical instrument compared to the size of the entire page

Three main research issues distinguish different Transfer Learning methods: 1) What to transfer? 2) How to transfer? When to transfer? [116]. Given the assumption that the source dataset is related to the target dataset, most of the focus remains on the How to transfer question. Moreover, we notice two main types of TL methods, intrusive and non-intrusive. Intrusive TL provides the best results overall and requires making hypotheses about the classes, the data, and especially the model’s architecture. Intrusive TL is very successful on image classification tasks since there is only one main architecture involved, which is Convolutional Neural Networks (CNN). However, it cannot be applied to more complex tasks such as Object Detection, which uses several model architectures and training procedures. The second type is Non-Intrusive TL, which provides worse results than intrusive TL. Nevertheless, it makes far fewer assumptions about the data and the task making it more applicable to several scenarios, ranging from Clustering to Object Detection and Segmentation. Non-intrusive TL is widely used in industrial applications because it is easily reproduced and provides more stable results.

Our method changes and adapts itself to different model architectures, namely Detectron2 and different YOLO versions. Our method’s strength comes from its simplicity and reduction of the model’s assumptions. Our method treats the object detection model as a black box with no Even though YOLO and Detectron architectures are different, YOLO is a one-stage detector, whereas Detectron first finds regions of interest and then classifies them in two separate steps. This difference allows YOLO to provide more real-time results even on simpler hardware, but it decreases the performance compared to Detectron, which was fixed in the later versions of YOLO.

The rest of the sections present the following: Section 2 provides an overview of the main research advances in Transfer Learning, Object detection and Medieval Manuscript Studies. Section 3 presents our model architecture and our dataset with its characteristics. Details of our experimental setup, and empirical results are shown in Section 4. Finally, Section 5 wraps up the chapter.

3.2 Method

This section describes our proposed method for Transfer Learning. Our method assumes the existence of two datasets that share the same classes either fully or partially, and one of them has a lower number of samples than the other. The goal is to improve the performance of object detection models on the smaller dataset. We refer to the big dataset as the source dataset X_s and the small dataset as the target dataset X_t . In the baselines and the results, we will refer to the Original Dataset X_o , a separate dataset used for object detection model initialization in general and not related to our source dataset annotations. The samples of both

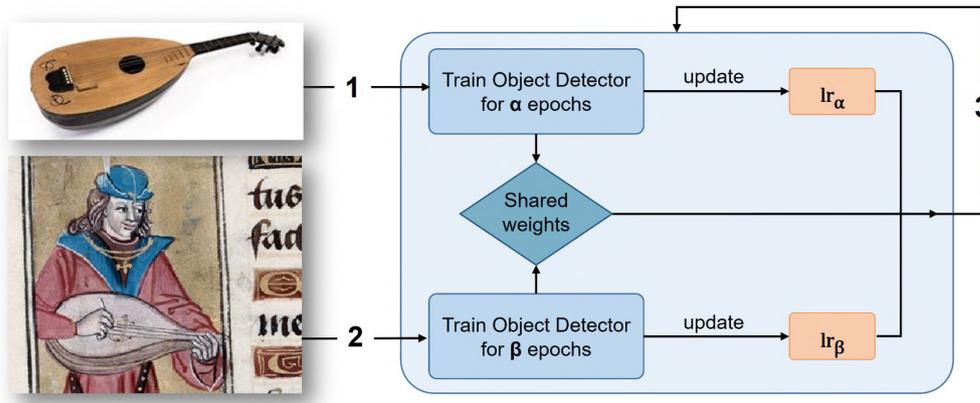


Figure 3.2: Dual Training for Transfer Learning method. The images are extracted from MIMO dataset (Source) and Musiconis (Target).

the source and target dataset can have different sizes since most object detection models are either flexible size.

It is essential to notice that the source and target images come from a different marginal distribution. This difference is at the core of the difficulties of performing normal Transfer Learning. We train the Object Detection model using both datasets at delayed times on two repetitive steps. The first step is to train on the source data, and the second is to train on the target dataset. Each dataset has its learning rate and its learning rate scheduler, but both share most of the model layers except for the last classification layers.

The source dataset starts with a big learning rate and then gradually decreases it as we advance in the iterations. In contrast, the target dataset starts with a low learning rate that gradually increases and goes back down in the middle of the training. Thus, the model initially learns most from the source dataset and then gradually focuses on the target dataset.

3.2.1 Baselines

We compare our method against several baselines to prove its efficiency. The baselines we chose are considered the best practice or the direct method of performing the object detection task.

Transfer Learning

The first baseline uses a model trained on a large dataset and fine-tuning it on a target dataset directly. This method is the most common form of training an object detection system. The idea behind this is to leverage a large number of

images and annotations in an external dataset to train the feature extractors of the model to recognize and distinguish a variety of classes. This method is not only widely used in object detection but also in image classification and natural language processing.

Dual Knowledge Transfer

Basic Transfer Learning suffers from one big loophole: the change in classes, object sizes, and image distributions between the original and the target datasets. This drawback is a widespread problem when working with historical datasets. That is why a new method for Transfer Learning appeared, which first trains the model on the original dataset. For example, the MS COCO 2020 Detection dataset or Google AI’s OpenImages. Using the source dataset to fine-tune the model the first time allows getting closer to the target dataset classes or its style of images. The next step is a fine-tuning on the target dataset to go much smoother and provide better results. The key difference is using three datasets instead of two, which allows the model to provide better performances on the target dataset.

3.2.2 Dual Training for Transfer Learning

Our method’s strength lies in its non-intrusive approach to improving the results of Transfer Learning and Dual Knowledge Transfer. Our method assumes that the object detection model is a black box that can take a learning rate and a number of epochs as input. Thus, it is more applicable to all object detection models based on deep learning. Fig 3.2 describes the structure of our method. We begin by initializing the weights of the model using a pre-trained model on the Original dataset X_O . We also set the learning rate for the source dataset (lr_α) and the learning rate for the target dataset (lr_β) to keep the initialization of $lr_\alpha^0 > lr_\beta^0$, allowing the model to focus mostly in the start on the source dataset. Our first step begins by training the model on the source data for α epochs and decreasing lr_α following this formula:

$$lr_{epoch} = lr_0 * exp(-k * epoch); k = ln(\frac{lr_{\beta 0}}{lr_{\alpha 0}}) * \frac{-1}{E_s} \quad (3.1)$$

where k is usually set to a specific value that allows the two learning rates to switch values in the middle on training, but it can be used as a hyper-parameter. The second step is to resume the training of the model’s weights using but with separate classification layers for the target dataset. We train on the target dataset for β , such that $\alpha > \beta$ since the number of pictures in the target dataset is assumed to be smaller than the size of the source dataset. We update the lr_β after every step following algorithm 5

Algorithm 1 : Upgrading the Learning rate for the target data

Input : $epoch$ — Current epoch
 lr_0 — Initial learning rate
 d — Decay parameter
 k — Increase parameter
 E_s — Switching epoch.

Output : lr_{epoch} — Current learning rate for the epoch

if $epoch < E_s$ **then**

In the first part of the training we increase the value of the lr;

$inc_{exp} = \exp(-k * epoch);$

$lr_{epoch} = lr_0 * (1 + inc_{exp}) ;$

else

In the second part of the training we decrease the value of the lr from the maximum.;

$inc_{max} = \exp(-k * E_s);$

$lr_{maximum} = lr_0 * (1 + inc_{max});$

$dec_{exp} = \exp(-k * E_s);$

$lr_{epoch} = lr_{maximum} * dec_{exp};$

end

return lr_{epoch}



Figure 3.3: Four examples of vielles. (a) Stone statue in Musiconis database. (b) Manuscript in Musiconis database. (c) Painting in Vihuelas database. (d) Photograph in MIMO database.

After we finish step 1 and 2 of our system, we iterate over and over again until we reach convergence or one of the stopping criteria has been fulfilled. We decide convergence by having a similar f1-score for both datasets, which exceeds a certain threshold. In the case where the convergence criteria are not satisfied, the system stops at one of the following criteria:

1. The max number of iterations was exceeded.
2. The difference between the f1-score of the two datasets keeps increasing for 5 consecutive steps.
3. The f1-score of the source dataset decreases for 5 consecutive steps.

These stopping criteria allow the model to stop running at the best iteration to provide the best results that satisfy the domain independence and category informative characteristics that we look for in a model.

3.3 Datasets & Challenges

To test our proposal, we selected three datasets of medieval artworks containing musical instruments: i) AnnMusiconis database²; ii) AnnVihuelas database³; and iii) MIMO Database⁴. The choice of these databases was to collect artworks from different periods, styles, and supporting materials:

²<http://musiconis.huma-num.fr/fr/>

³<https://vihuelagriffiths.com/>

⁴<https://mimo-international.com/MIMO/>

Algorithm 2 : Dual Training for Transfer Learning

Input : $lr_{\alpha 0}$ — Initial learning rate for the source dataset.
 $lr_{\beta 0}$ — Initial learning rate for the target dataset.
 α — the number of epochs to train the model on the target dataset every step.
 β — the number of epochs to train the model on the source dataset every step.
 E_s — Switching epoch.

Output : W_{target} — The weights of the target model.

$lr_{\alpha} = lr_{\alpha 0}$;

$lr_{\beta} = lr_{\beta 0}$;

$k = \ln\left(\frac{lr_{\beta 0}}{lr_{\alpha 0}}\right) * \frac{-1}{E_s}$;

for $epoch \leftarrow 1$ to $2 * E_s$ **do**

for $epoch_{\alpha} \leftarrow 1$ to α **do**

 Sample a batch of images from the source domain;

 Train the object detection model using lr_{α} ;

 Update lr_{α} using the formula [3.1](#) and k ;

end

for $epoch_{\beta} \leftarrow 1$ to β **do**

 Sample a batch of images from the target domain;

 Train the object detection model using lr_{β} ;

 Update lr_{β} using the algorithm [5](#);

end

end

return lr_{epoch}

- 1. Large historical periods:** These collections include artworks from 11 different centuries. First, the AnnMusiconis database contains images mainly of stone statues from the 9th century to the 17th century. Second, the AnnVihuelas database contains artworks from the early modern period (15th to 17th centuries).
- 2. Several supporting materials:** These datasets include musical instruments in manuscripts, paintings, stained glasses, embroideries, photographs, stone and ivory sculptures. Thus, we will evaluate our models and compare the results obtained according to the material used to create the artworks.
- 3. Variations in representation styles:** . The datasets contain musical instruments' variations due to different artists' painting or sculpting styles. As we mentioned before, the artist who creates the artwork is not necessarily a musician, so the instrument drawn or sculpted may not correspond to the real instrument's characteristics (for example, a missing string, a shorter neck, or a wider soundboard).
- 4.- Different conservation states:** Since we focus on medieval artworks, some may suffer from corrosion and are often damaged. Others have been broken or shattered, making their instruments difficult or harder to recognize.
- 5. Rich and broad application domain:** There are thousands of musical instruments throughout history worldwide. Due to the large number and variability of musical instruments in these databases (more than 65k records), we decided to identify only chordophones. A chordophone is a musical instrument that produces sound from vibrating strings, such as harps, lyres, or lutes.
- 6. Different object sizes:** Object detection models work perfectly on images where the object sizes are big and the objects are not crowded together. Fig. [3.4](#) shows the kernel density estimation of the empirical distributions of the different datasets.

To get a glance on our MIMO, AnnMusiconis, AnnVihuelas (MMV) dataset and understand their differences, we present Fig. [3.3](#) which depicts four images of vielles that vary in color, shape, size, orientation, and perspective. The Vielle was a chordophone widely used in the Middle Ages, currently considered an ancestor of the modern violin.

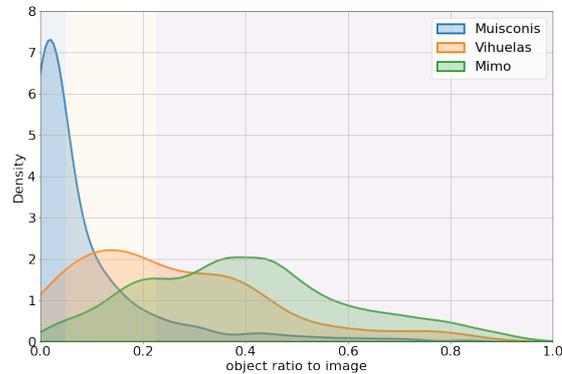


Figure 3.4: The kernel density estimation of empirical probability density function of the object area to image size ratios for MMV datasets. The sections represent the range we chose for deciding small, medium, large sizes for evaluation.

Table 3.1: Per-class Performance map comparison between several Object Detection models on the task of Transfer Learning. TL refers to vanilla Transfer Learning, DKT refers to Dual Knowledge Transfer and DTTL refers to our method. The experiments use MIMO dataset as a source dataset, and either AnnMusiconis or AnnVihuelas as the target.

Target Dataset		MIMO AnnMusiconis				AnnVihuelas			
OB backbone	TL Method	Precision	RECALL	f1-score	map	Precision	RECALL	f1-score	map
Detectron	TL	56.8	47.1	51.49	34.40	73.9	56.3	63.91	43.412
	DKT	59.4	48.51	53.40	35.63	72.41	59.0	65.02	42.95
	DTTL	61.35	51.8	56.17	36.09	73.36	61.84	67.11	43.23
YOLO v4 csp	TL	58.49	49.29	53.49	45.07	72.15	58.57	64.65	46.01
	DKT	58.39	52.31	55.18	46.39	70.81	60.64	65.33	48.14
	DTTL	64.73	53.7	58.71	48.14	74.22	62.18	67.69	52.03
YOLO v4	TL	55.42	35.91	43.58	32.22	54.29	52.41	53.33	43.47
	DKT	57.34	41.45	48.11	33.58	58.94	55.14	56.97	43.81
	DTTL	59.76	46.68	52.42	34.02	67.49	59.07	63.00	45.98
PP-YOLO	TL	54.74	58.29	56.45	42.45	70.95	57.62	63.59	40.92
	DKT	57.21	61.01	59.04	44.28	73.47	59.12	65.51	41.91
	DTTL	59.78	60.79	60.28	47.85	75.12	60.34	66.92	43.99
YOLO v5 - m	TL	52.33	48.17	50.16	32.72	58.29	53.45	55.76	38.07
	DKT	55.54	48.41	51.73	33.15	64.74	51.79	57.54	40.06
	DTTL	58.79	51.72	55.03	35.41	66.57	54.68	60.05	41.53

Table 3.2: Per-class Performance map comparison between several Object Detection models on Transfer Learning’s tasks. TL refers to vanilla Transfer Learning, DKT refers to Dual Knowledge Transfer, and DTTL refers to our method. The experiments use the MIMO dataset as a source dataset and either AnnMusiconis or AnnVihuelas as the target.

Target Dataset		AnnMusiconis						AnnVihuelas		
OB backbone	TL Method	viele	Archet	Cithare	Harpe	Luth	Lyre	viele	Archet	Luth
Detectron	TL	30.256	25.894	23.90	35.922	41.63	48.82	30.297	60	39.93
	DKT	32.12	28.46	22.58	32.85	48.69	49.12	32.79	53.4	42.66
	DTTL	40.27	30.08	23.48	31.1	42.55	48.99	36.32	52.26	41.1
YOLO v4 csp	TL	64.54	32.71	48.37	57.53	34.78	32.54	25.64	55.01	57.4
	DKT	62.84	32.57	51.99	54.23	39.49	37.26	29.15	56.71	58.57
	DTTL	60.54	38.12	48.29	57.93	43.28	40.72	37.96	55.8	62.25
YOLO v4	TL	44.44	33.42	20.68	41.4	24.17	29.22	26.93	50.82	52.67
	DKT	42.44	35.13	25.48	43.35	27.07	28.01	27.45	52.38	51.6
	DTTL	45.26	36.21	32.1	41.11	17.01	32.47	29.09	54.48	54.38
PP-YOLO	TL	61.18	42.17	48.33	58.29	28.09	16.69	26.81	53.88	42.07
	DKT	58.81	45.63	44.52	55.86	38.29	22.58	31.52	50.93	43.3
	DTTL	60.28	48.15	49.24	52.99	47.12	29.37	34.67	52.01	45.29
YOLO v5 - m	TL	42.17	38.33	35.62	28.95	30.28	21.02	38.25	43.52	32.45
	DKT	43.01	34.29	37.24	29.03	29.85	25.46	36.13	48.56	35.48
	DTTL	45.57	32.48	41.19	35.2	32.04	26.03	37.04	48.02	39.54

3.4 Results

This section provides a detailed comparison between the vanilla method for Transfer Learning, the Dual Knowledge Transfer, and our non-intrusive Transfer Learning method for object detection named Dual Training for Transfer Learning. As object detection is a central area of interest in the computer vision community, several models appear every month and lack a proper comparison. We provide a detailed comparison between them based on f1-score, mAP, training time, inference time, and model size. The models that we evaluated are: YOLOv4 (original, tiny, scaled), YOLOv5 (s,m,l,x), PP-YOLO, Detectron 2 (Faster-RCNN). We present our results in two sections. First, in Section 3.4.1 we present the evaluation based on global metrics for performance and computations. Second, in section 3.4.2 we provide a per-class performance improvement comparison to present the effect of different models and methods for Transfer Learning on unbalanced datasets.

3.4.1 Global Performance Evaluation

Following the steps described in Algorithm 2 we evaluated our method on the task of Transfer Learning for object detection on our novel historical music datasets. The goal is to improve the performance of several Object detection Methods on our

Table 3.3: Hyper-parameters for the transfer learning experiments on musical instruments object detection for Table 4.1 and Table 3.2. The annotations are described in Algo 2.

Hyper-paramter	lr_α	lr_β	α	β	E_s
Detectron	0.00025	0.00005	15	5	300
YOLO v4 csp	0.001	0.0005	10	5	150
YOLO v4	0.00261	0.0005	15	5	125
PP-YOLO	0.01	0.001	10	5	500
YOLO v5 - m	0.0001	0.00003	20	5	300

target dataset (namely AnnMusiconis and AnnVihuelas) to predict the location and class of several musical instruments. Table 4.1 presents the results of our method against vanilla TL and Dual TL. The metrics used for evaluation are Precision (the number of positive class predictions that belong to the positive class), Recall (the number of positive class predictions made out of all positive examples in the dataset), F1-score (A balance between precision and recall), and mean Average Precision score@[.50:.05:.95] (the average AP for Intersection over Union from 0.5 to 0.95 with a step size of 0.05.). The different inference threshold used for precision, recall and f1-score calculations are reported along with the hyper parameters of each model in table 3.3

Table 4.1 shows that for vanilla Transfer Learning the best models to use are PP-YOLO and YOLO v4 scaled P7 models. PP-YOLO has a higher f1-score with 0.5829, an increase of 0.02961 over YOLO-scaled, whereas the latter provides an increase of 0.0262 over PP-YOLO with 0.45 on map score. Besides, both models remain at the top of the list when combined with Dual Knowledge Transfer, but their improvement over vanilla TL is small compared to the results with our method (DTTL). Table 4.1 shows that our TL method improves results for all models compared to DKT or TL. Our method also increases the f1-score of object detection models with an average of 4.79 and a maximum increase of 8.83 for the YOLOv4 model.

3.4.2 Per-class Performance Evaluation

To better understand and evaluate our method, we show in Table 3.2 the per class-map scores on the different target datasets. The experiments are the same as before, and the train/valid/test splits are the same across all models and all TL methods, and the hyper parameters of the method are reported in Table 3.3. The results show that the map scores correlate heavily with the distribution of samples per class, meaning classes with higher sample count have better map scores on

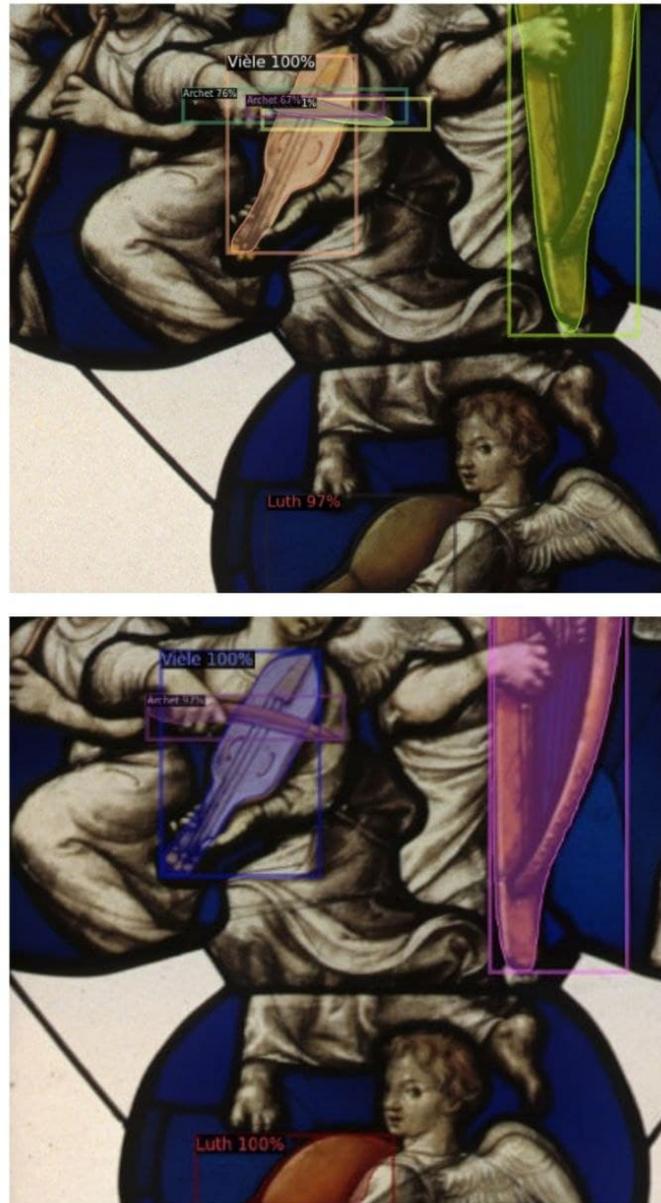


Figure 3.5: Comparison between results of our method DTTL (on bottom) vs traditional transfer learning method DKT (on top).

average than those with a small sample count. For example, the Lyre and the Luth map scores of strong models such as YOLOv4 csp and PP YOLO are lower than the same models' map scores on classes such as the viele and the harpe.

As mentioned before, the best two models overall are YOLOv4 csp and PP-YOLO with YOLOv4 csp slightly taking the lead. These two models provide overall good results for all classes of both datasets (AnnMusiconis and AnnVihuelas). Their performance increases when we use the dual knowledge transfer method over simple transfer learning, especially on classes with a smaller sample count, such as Viele for the AnnVihuelas dataset, and Luth/Lyre for the AnnMusiconis dataset. Nevertheless, our transfer learning method outperforms the DKT method and provide a larger improvement on classes with smaller sample counts. Taking the example of YOLOv4 csp on the Luth class of AnnMusiconis, we can see that the normal TL method gives a 34.78 map score whereas DKT improves on this with 39.49 providing an increase of 4.71, whereas our method gives 43.28 with an increase of 8.5, which is almost the double.

We can also confirm the improvements provided by our method visually by analyzing the results' annotations in the images. For example, Fig. 3.5 shows an image of angels holding several musical instruments (a harpe, a luth, and a viele played with a bow). Our method provides more certain results with less noise, which can be verified on the bow example. The DKT method with Detectron detects several overlapping instances of the viele and none of them match the correct form of the instrument (first image on top of Fig. 3.5); whereas our method provides only one precise detection of the bow with higher probability (second image at the bottom of Fig. 3.5).

3.5 Conclusion

In this chapter, we presented a new transfer learning method integrates easily with all state of the art models for object detection and provides a significant performance increase over other transfer learning methods. Our transfer learning method is an improvement in the field of medieval manuscript studies because it allows us to train models and improve their results ever so slightly so they become useful to us in future applications. Even though we build the algorithm with the assumption of lack of target data, our method still requires a bare minimum of images to train on. This drawback lead us to think of new and different methods for building models, even in situations where data is absolutely sparse. We tackle this drawback in the next chapter.

Chapter 4

Few Shot Object Detection

The methods and some results presented in this chapter were published in:

Ibrahim, Bekkouch Imad Eddine, Victoria Eyharabide, Valérie Le Page, and Frédéric Billiet. 2022. "Few-Shot Object Detection: Application to Medieval Musicological Studies" *Journal of Imaging* 8, no. 2: 18. <https://doi.org/10.3390/jimaging8020018> [72].

In this chapter, we explore the field of few-shot object detection by presenting a new non-intrusive method relying on bi-training of object detection models. The core of the idea is to train the object detector sub-part of the model on the entire dataset and then follow it by tuning the classifier/bounding box regressor part of the model by assigning a higher sample weight on the novel classes. This method is tested on our dataset MMSD for medieval singing.

4.1 Introduction

The analysis of medieval vocal practices is an essential issue for musicologists and performers. However, medieval singing as a musical performance has been explored much less than other musical instrumental performances [15]. This is because medieval musical instruments are studied mainly by analyzing images of artworks in which these instruments are represented [13]. However, since the human vocal cords cannot be displayed explicitly, it is harder to identify whether a person or group is singing or not.

Even though there are numerous descriptions of musicians and singers in medieval chronicles and tales, illuminated manuscripts are the principal source for musical iconography. Illuminations often depict very complex situations in a tiny space. Artists often wished to concentrate much more information in a small illumination than would be contained within that scene in real life. However, studying a large corpus of images allows musicologists to detect repeated patterns and shed

light on previously unknown medieval vocal practices across different periods and regions. The discovered patterns could enable performers wishing to perform repertoires to better understand the organization of singers, the environment, and the setting of the songs according to the period and genre considered. For example, considering the architectural modifications over the centuries, better choices regarding locations and musicians could be made to recreate acoustics as close as possible to the original music scene. Therefore, our objective is to find singing performances in images from medieval artworks. More precisely, we will detect medieval images containing persons in solo or group-singing situations, whether accompanied or not by musical instruments. The final objective for musicologists is to better understand the physical postures of singers, their relationship, and their location inside the building.

Since the human voice is not a visible musical instrument, it is necessary to define possible objects in the images that may suggest the presence of singing performances. Therefore, we propose identifying characters who have their mouths open, perhaps with features linked to the vocal utterance (such as declamation or singing; see Figure 4.1a¹). However, having the mouth open is not a sufficient condition to determine that a person is singing. The context or environment in which these singers are performing is vital to understanding the musical scene.

The scene's context should be analyzed to detect additional clues such as a book held in the hands or knees, or placed on a lectern (Figure 4.1b², an unfolded phylactery (and if visible, the text on the phylactery, Figure 4.1c³), or some musical notation (Figure 4.1d⁴). Moreover, some gestures such as the hand placed on the shoulder, the movement of the pulse (to set the tempo and anchor the rhythm), or a finger pointing to the musical score may also evoke singing performances. Musicologists also analyze the texts embellished by the miniatures or illuminations containing singing performances. This research deliberately sets aside animals, hybrids, and monsters to concentrate only on clerics, laity, children, and angels.

When dealing with small datasets (such as those for medieval studies) that are challenging even for typical transfer learning methods, few-shot image classification is a possible solution [79]. The technique of few-shot learning [168] in computer vision has progressed drastically over the last years, mainly due to the advances in transfer learning techniques such as meta-learning [160]. Such advances have

¹(BnF ms. fr. 166 f. 121v: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b105325870/f256/1500,750,1000,1300/full/0/native.jpg>, accessed on 1 November 2021))

²(BnF ms. fr. 166 f. 115: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b105325870/f243/2799,3000,1000,1200/full/0/native.jpg>, accessed on 1 November 2021))

³(BnF. NAL 104, f. 50r: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b10023007f/f106/50,50,2000,2300/full/0/native.jpg>, accessed on 1 November 2021))

⁴(BnF ms. fr. 166 f. 126v: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b105325870/f266/3199,3100,1000,1200/full/0/native.jpg>, accessed on 1 November 2021))



(a) Book in their hand



(b) Book placed on a lectern



(c) Unfolded phylactery



(d) Book with musical notation

Figure 4.1: Examples of medieval singing illuminations.

provided great results for basic image classification tasks. Our datasets in the field of cultural heritage suffer from a great deal of challenges, as described in [13], but the main challenges are the lack of samples in specific classes due to the loss of historical artifacts or the difficulty of finding such samples in the vast collections provided by different museums. Unlike image classification [7], few-shot object detection has received far less attention in the past and is still a growing field. The main difference between image classification and object detection is that the model is required to detect the location of the classified objects from a possible set of millions of potential locations. This additional sub-task makes the object detection task even harder to perform in scenarios where annotated data are sparse.

In this chapter, we present two main contributions, (i) a novel technique for performing few-shot object detection based on bi-stage training, in which the first stage tries to improve on the object localization process for the new classes and the second stage aims to improve the image classification and fine tuning of the pre-located coordinates; and (ii) a benchmark for three main models in the field of object detection, which are Yolov4 [18], Faster RCNN [125], and SWIN Transformers [102]. We chose these three architectures because they represent the leading representatives of their family trees of architectures, which are the RCNN family [56, 64, 57], the YOLO family [123], and the Visual Transformers family [171].

The chapter is organized as follows: Section 4.2 presents our novel and simple method for few-shot object detection. The empirical benchmark of our algorithm is shown in Section 4.3. Finally, Section 4.4 summarizes the contributions of the chapter.

4.2 Methodology

In this section we describe our novel method for few-shot object detection based on a bi-training approach. We start by first describing the context of few-shot object detection and the type of input we work with; then we describe in detail the different steps and loss functions for our bi-training method. We finish this section with a description of the contributions provided in this chapter.

We use the same description of few-shot object detection as the settings introduced in [79]. We have a set of base classes C_b , which contain a sufficient number of samples (sufficiency depends on the model and the pretraining and the method used for training, which we will investigate in the results section) and a set of novel classes C_n that have a low representation in the dataset, with only K objects per class (where K is small number, usually around 10; we investigate different values of K in the results section). For the object detection task we need $D = (x, y), x \in X, y \in Y$ such that x is the input image and $y = (c_i, l_i), i = 1, \dots, N$ is the set of annotations per image that is made with the

bounding box coordinates and the class of each object such that the set of classes is the union of base classes and novel classes. For our dataset, we use a different ratio of novel/base classes and different thresholds for the definition of a novel and base class.

To evaluate our method for few-shot object detection, we used a test set that contained a combination of both novel and base classes, with the final goal of optimizing the performance of the model on both sets, which we quantified using the mean average precision (mAP) metric, which combines the results on all classes. We describe our bi-training few-shot object detection (BiT) method in this section. We demonstrate that our method is model-agnostic and can work with a variety of model architectures, allowing it to be used on any pretrained object detector, fine-tuning it for enhanced performance. We chose to use the faster RCNN approach with a region proposal network (RPN) backbone as the representative of region-based convolutional neural networks, and to use Yolov4 as the representative of the You Only Look Once family and the Swin T model as a representative of the vision transformer family. Intuitively, our method aims at treating the object detector as a black box but makes a few assumptions about it. The first assumption is the existence of an object proposal sub-network (for most two-stage models this exists easily but for Yolo it is represented as a model with only the “objectness score calculations” without performing the object classification, making it a class-agnostic sub-model). The second assumption is the existence of an object classifier, which is the case for all of our models. Our method operates on two training steps described in Algorithm 7, which are:

Total Model Improvement

The first step in our method is to fine-tune the object detector (not the classifier) on the whole dataset, not only the base classes, to make sure the model is able to propose objects for classification for all classes, especially if the novel classes are not very similar in shape and style to the base classes. This step can be applied to all object detection models since they are all either two-step models or YOLO-based, where the classification and object proposal happen in the same step, and this can be achieved by using a weighted combination of both losses, where the weight of the classification at this step is 0. The joint loss would look like this:

$$\mathcal{L} = \mathcal{L}_{rpn} + 0 \times \mathcal{L}_{cls} + \mathcal{L}_{loc}, \quad (4.1)$$

such that \mathcal{L}_{rpn} represents the object proposal loss function applied on the RPN (or the object score cross entropy loss for YOLO-based models), which mainly used to refine the anchors without touching the feature extractor (the backbone remains fixed as our dataset does not have enough samples to effectively retrain the

whole model). \mathcal{L}_{cls} is the object classification cross entropy loss and in this stage we do not train it; we try to ignore its effect because the object proposal step at this stage still is not good enough to propose enough samples of the novel classes, leading to an even bigger class imbalance problem for the classifier. Finally, \mathcal{L}_{loc} is the smoothed L1 loss used to train the box regressor.

Classifier Fine-Tuning

The second step of our model treats the object detector as a whole as a two-step model, in which the first step is the object proposal and the second is the classification and bounding box regression. In our first step we fine-tuned the object proposal model and now, for the fine-tuning of the classifier and bounding box regressor, we will treat it as we would another CNN model used for classification. This simplification works very well on all object detector models, regardless of whether they are one-step models (Yolo) or two-step models (RCNN). The fine-tuning is targeted only at the last layer of the bounding box regressor and the classifier (if they are separate layers, as in the RCNN family, they are fine tuned separately, and if they are in the same layer, as in YOLO, they are done together). We fine-tune both base classes and novel classes but we assign a higher sample weight to the novel samples, forcing the model to perform better with the novel samples. We decrease the learning rate in this stage compared to the first stage, allowing the model to train slower and not to change drastically in order to fit the novel classes and abandon all results on the base classes.

4.3 Results

In this section, we conduct extensive benchmarking of several object detectors on the task of few-shot object detection based on two approaches. The first one aims at measuring the influence of reducing the overall number of samples for both base classes and novel classes on the performance of the model in total. The second experiment was conducted to see how the change in the number of samples of the lowest novel class influenced its own average precision. In this chapter, we focus only on deep learning models instead of machine learning techniques, mainly because our dataset contains a large variety in forms shapes, sizes and artistic representations within the same group of objects. Nevertheless, our dataset has a small sample count for each class, making it a very difficult task for machine learning models which do not leverage pre-training and transfer learning. For the sake of a better evaluation, we trained a histogram of oriented gradients, followed by a support vector machine classifier and a sliding bag of visual words. The results for the Bag Of Visual Words BOVW model were almost zero regardless of the fine

Algorithm 3 : Bi-Training Few-Shot Object Detection

Input : X — Training images.
 Y — Training labels.
 W_n — Sample weight multiplier for the novel classes.
 lr_1 — First step learning rate.
 lr_2 — Second step learning rate.

Output : θ^O — Weights of the object detector

```

// Training the object detector sub-part of the model.
for  $i \leftarrow 1$  to  $epochs$  do
  for  $j \leftarrow 1$  to  $nb\_batches$  do
    | Sample a batch of images regardless of source class
    |  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ ;
    |  $\theta^O = \theta^O - lr_1 \frac{\partial L}{\partial \theta^O}$  Equation (4.1) ;
  end
end

// Sample weight calculations and classifier fine-tuning
 $\xi_w = norm([1 \text{ if } y_i \in C_b \text{ else } W_n \text{ for } i \in [1 \dots N]])$ 
for  $i \leftarrow 1$  to  $epochs$  do
  | Sample a batch of images for both domains and their labels and weights
  | from  $\xi_w (x^b, y^b, w^b), (x^n, y^n, w^n)$ ;
  | Update  $\theta^O$  by deriving  $\mathcal{L}_{rpm} + \mathcal{L}_{cls} + \mathcal{L}_{loc}$ ;
end
return  $\theta^O$ 

```

tuning, but the Histogram of Oriented Gradients HOG model had a high detection rate (74/138–80% training, 20% testing) but a very low precision since the model confused many of the other parts of the image as an object, dropping the f1-score to 0.239. The threshold of the intersection over union used for the experiment was 50% to increase the model’s results.

We used multiple models for benchmarking, mainly the yolov4 (m/s), mask RCNN (faster RCNN/Mask RCNN), and ViT (Swin-t/ViT) models. Each one of these models was trained fully on the dataset to provide us with the upper bound estimation of the baseline for few-shot object detection, which expresses the full object detection capabilities of the model. We also use a lower bound baseline to prove the effectiveness of our method for few-shot object detection by training the models on the few-shot data directly, without adding any emphasis on the novel classes. For the sake of the experiments in the rest of this section, we selected two base classes, which were livre (book), and phylactère (phylactery), mainly because they were highly represented in our dataset, with 338 objects for book and 204 for phylactery. The novel classes are represented in the rest of the object classes, specifically: lutrin (lectern), autel (altar), feuillet (leaflet), texte chanté (sung text), which were much less represented in the dataset, with lectern having 87 samples, whereas the others had between 20 to 30 samples each. Although many methods have been proposed for hyper-parameter tuning, such as Bayesian-optimized bidirectional LSTM [81] and Google Vizier [58], we chose to hyper-parameter-tune our models using the default hyper-parameters of each backbone or using grid-search cross validation, as described in hyperparameter optimization [45], with $n_{epochs} = 2000 \times n_c$ and $n_c = n_b + n_n$ as the total numbers of classes (base and noval), whereas the learning rates, for example, were set to [0.0001, 0.00025, 0.001, 0.01, 0.00001], and through grid search CV we found that we obtained the most optimal results by using $lr_1 = 0.00025$ and $lr_2 = 0.00001$.

4.3.1 Global Few-Shot Object Detection Benchmark

We provide the average AP50 of the models on the whole dataset of medieval singing images with different distribution percentages of data (100%, 80%, 50%). These percentages applied to each object class individually to keep the same ratio of classes in the dataset. The goal of this evaluation was to see how the model’s performances changes on novel classes and on base classes.

The data were split into training and testing data, following different ratios, 90% 10% for the base classes and 60% 40% for the novel classes, allowing a relatively good amount of objects for novel classes to evaluate and extract meaningful information and ranking between models. As the number of testing samples for the novel classes still remains too small to be statically significant to extract useful interpretations, we report the median results of the models over five repetitions



Figure 4.2: Inference of the Mask RCNN architecture for few-shot instance segmentation on our medieval singing dataset.

Table 4.1: Average Precision evaluation of different state-of-the-art models for object detection using our newly proposed dataset. LB refers to the lower bound baseline, which is transfer learning with the same data, and UB refers to upper bound baseline, which is a transfer with the full dataset.

		Percentage									
		100% UB		80% Ours		80% LB		50% Ours		50% LB	
		Base	Novel	Base	Novel	Base	Novel	Base	Novel	Base	Novel
Yolov4	s	83.62	81.86	61.88	62.24	50.24	55.66	54.78	48.57	52.85	51.30
	m	72.86	77.19	65.27	58.87	58.28	58.66	36.94	52.54	33.76	36.77
RCNN	Faster	68.90	72.04	61.55	58.44	52.39	54.54	46.77	47.19	42.04	37.08
	Mask	79.13	77.472	71.31	62.49	59.77	62.17	55.58	57.03	49.15	54.03
ViT	ViT	57.13	54.96	45.27	42.72	36.84	50.12	33.05	35.55	28.53	23.26
	Swin-t	72.08	57.49	65.18	45.35	48.75	45.25	44.30	46.77	38.06	44.49

of the same experiment but with random train test splits, allowing for more stable estimations of the performances of each model and for the effectiveness of our few-shot object detection method, and Figure 4.2 shows some inference results on our dataset for this .

Table 4.1 shows the mean average precision values of the different models on different versions of our dataset. We can see from the results of the upper boundary with 100% data that the order of models was as follows—yolov4-s was the best for our dataset, providing good performance for both base and novel classes, followed by mask RCNN and yolov4-m, whereas the Swin-t model provided good results on the base classes and average results for the novel classes. The worst model for our dataset was ViT, which requires a lot of data to train a good model and is thus

Table 4.2: Average precision evaluation of different state-of-the-art models for object detection using our newly proposed dataset. LB refers to the lower bound baseline, which is transfer learning with the same data, and UB refers to the upper bound baseline, which is a transfer with the full dataset.

		Lowest Novel Count				
		Full UB	10 Ours	10 LB	5 Ours	5 LB
Yolov4	s	76.15	59.78	52.13	49.71	43.79
	m	71.89	53.17	54.40	48.86	31.82
RCNN	Faster	68.17	52.85	53.35	42.08	38.57
	Mask	72.65	57.92	57.06	51.14	48.60
ViT	ViT	52.69	46.18	41.92	35.97	24.05
	Swin-t	53.61	47.99	40.20	41.88	39.28

not suitable for few-shot object detection tasks and cultural heritage applications in general. We can also clearly see that our model always improves over the lower boundary of transfer learning when it is used in isolation for all models and all versions of the dataset, proving the effectiveness of our simple yet effective method. Our model works best for the Mask RCNN model, which is the most compatible with our idea and assumptions, showing a performance for 80% similar to 100% and for 50% similar to the LB for 80%.

4.3.2 Worst-Case Few-Shot Object Detection Benchmark

In this section we evaluated the performance of the different object detection models on the task of few-shot object detection while changing the number of samples in the smallest class and keeping the others full. This gives us an estimation of the worst-case performance for a specific class by any of these different object detectors, allowing for a better benchmarking and better decision-making when choosing a model for different application areas.

Table 4.2 shows the results of the different models on the different states of our dataset, especially for the lowest represented class. We repeated each experiment five times and each time we randomly selected a class and set its instances for training to the chosen number for the experiment, and we report the mean average precision of the model over all the experiments.

The previously described setup allows us to obtain a good estimation of what would be the worst results obtained by each model in the task of few-shot object detection. We can see from the results that the ordering is still the same, with

yolov4-s still taking the lead, followed by Mask RCNN. Our method also improved the performances over the transfer learning baseline, showing the effectiveness of our method. We can also see that attention-based methods are very useful and are indeed the current state-of-the-art for object detection but they still require a lot more data to train a good model than all other previous methods such as Yolos and different versions of RCNN.

4.4 Conclusions

We have presented a new and simple few-shot object detection method integrates seamlessly with all state-of-the-art models for object detection, such as YOLO-based, RCNN-based, and attention-based methods, and provides a significant performance increase over traditional transfer learning methods, yet remains very limited in extreme cases where sample counts are very small. We also concluded that attention-based models are very powerful, but they require more training data, unlike models such as YOLOv4 s and YOLOv5 v6.

Our novel method of few shot object detection solves a major issue of drastic data sparsity and provides a bare minimum model in situations where previously it was impossible to even train a model. Yet, this is not enough for building a great model, as transfer learning techniques only touch the surface of the model by design due to their non-intrusive nature and the aim to generalize to different architectures.

Chapter 5

Auxiliary learning: Adversarial Domain Adaptation

The methods and some results presented in this chapter were published in two articles:

Bekkouch, Imad Eddine Ibrahim, Youssef Youssry, Rustam Gafarov, Adil Khan, and Asad Masood Khattak. 2019. "Triplet Loss Network for Unsupervised Domain Adaptation" *Algorithms* 12, no. 5: 96. <https://doi.org/10.3390/a12050096> [11].

Bekkouch, I.E.I., Constantin, N.D., Eyharabide, V., Billiet, F. (2022). Adversarial Domain Adaptation for Medieval Instrument Recognition. In: Arai, K. (eds) *Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems*, vol 295. Springer, Cham. https://doi.org/10.1007/978-3-030-82196-8_50 [13].

In this chapter, we will abandon the goals of non-intrusivity and make drastic changes to the architecture and loss functions of the models in order to improve their training. This is called domain adaptation and we approach it using the idea of Auxiliary learning. This method is tested on our datasets AnnMusiconis, AnnVihuelas and the famous MNIST-SVHN-USPS datasets.

5.1 Introduction

Machine learning (ML) has become part of our everyday lives, from ads on our phones to self-driving cars and smart-homes. Mainly because we now have an abundance of computational power and large datasets to train models for every task. There are three main types of ML which are supervised learning (Video Recognition [7], Diagnosis [12]), unsupervised learning (Domain Adaptation [11], outlier detection [179, 130, 71]) and semi-supervised learning (Speech analysis,

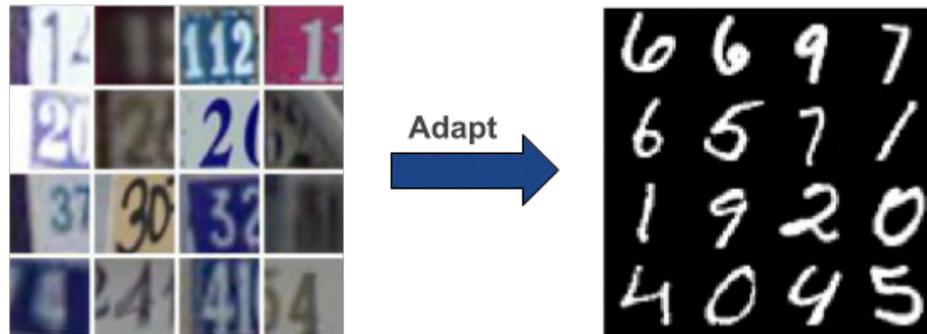


Figure 5.1: Case study of domain adaptation. Domain adaptation aims at closing the discrepancy gap between different datasets from different domains SVHN, i.e., street view house numbers and MNIST, i.e., hand written digits, while preserving good performance on a specific task, i.e., digits classification.

Spam Detection [30]) which are defined based on the availability of training annotations data. Supervised learning is usually the easiest and most advanced type given its predictability and stability compared to other types as it gives the required output and can be tailored to each task.

The primitive solution to the problem of domain gaps is to adapt the model for the new (or target) domain by retraining the model on the data from the target domain. However, the collection of new data and the retraining of the whole model can be difficult, expensive and even impossible. Hence a better approach is to store the knowledge learned in the primary domain and later transfer that knowledge to the target domain that shares the same tasks but could follow a different distribution. This can help in reducing the cost of data recollection and its labeling.

Let D^s and D^t be the source and target domains, respectively. Domain adaptation (DA), which is a sub-field of transductive transfer learning (TTL), aims to solve a problem in D^t , where data are hard to collect, using data from D^s . Both domains, usually, share the same tasks, i.e., $T^t = T^s$ but the marginal distributions of the inputs differ, i.e., $P(X^s) \neq P(X^t)$, as shown in Fig. 5.1. DA is usually achieved by learning a shared feature space, i.e., $Z^s = Z^t$ [116].

DA can be categorized as either closed-set or open-set. Closed-set DA is the case where the classes of D^t are the same as that of D^s . Our work belongs to closed-set DA. On the other hand, open-set DA handles the case where only a few classes are shared between the two domains, and the source or the target domain might contain more classes.

Similar to other machine learning tasks, DA can be split into supervised, unsupervised and semi-supervised depending on how much labeled data are available from D^t . For supervised domain adaptation (SDA) [112] and semi-supervised domain adaptation (SSDA) [181], the data are completely or partially labeled but it is not sufficient enough to train an accurate model for the target domain from scratch. In unsupervised domain adaptation (UDA) [22] [133] the target domain samples are completely unlabeled, which is useful in situations where the data collection process is easy but the data labeling process is time consuming. The extreme case of DA is when we don't have any access to the target data, and it is called domain generalization (DG). In DG, researchers have mainly used easy-to-collect datasets from different domains to make a model that can generalize well to unseen domains [38].

The focus of this chapter is on UDA which typically needs large amounts of target data specially in the case of deep unsupervised domain adaptation (DUDA). Although, the focus is on DUDA because of a wide variety of real world applications that it can solve, we use SDA as an upper bound to aim for because SDA typically outperforms UDA, and we will exploit this fact to make our model perform even better using a concept called pseudo-labeling [94]. In the DA literature, the early contributions considered sample re-weighting [124] as an attempt to outweigh the samples that almost shares the same distributions with the target domain D_T such as empirical risk minimization [17] and covariate shift [1] [70]. The weak point of this approach is that the support of both the source X_S and the target domain X_T may not be shared.

Most of the previous DUDA approaches aim at achieving two targets: (i) produce (or learn) feature vectors from the data from D^s that can be used by a classifier to get highly accurate class labels, and (ii) make the features of both D^s and D^t indistinguishable. Both Z. Ren et al in [126] and Lanqing Hu et al in [67] used generative models in different manners to achieve those targets. The former used GAN to reconstruct various property maps of the source images while keeping the features extracted of both domains similar by training a base encoder on the opposite loss of the discriminator. The latter work used duplex GAN architecture that can reconstruct the input images in both flavors, source and target, using the features extracted from the encoder. Next, a duplex discriminator is trained to distinguish the reconstructed images into source and target.

In this chapter we provide two methods for UDA, which are incremental improvements of the same core idea, defining a negative behavior as a loss function and training in an adversarial manner to reduce this negative behavior. Our models are motivated by the latter work, yet they are much simpler. More specifically, we show that the generative part is hard to train and is not necessary to obtain a domain adaptive model. By introducing novel loss functions, we show that our

models produce comparable results to the state-of-the-art model in a computationally efficient way.

Our method’s focus is on unsupervised domain adaptation where the source dataset does not contain labels as it is a less researched area and has several applications in historical and medieval datasets. Although this data is always annotated, they are small in size, making it hard to train models on them directly. Previous UDA methods aim at achieving two requirements for the shared latent space: (i) extract (or learn) a latent space representation from D^s and D^t that are class informative and useful for determining and separating the classes from each other, and (ii) Making the feature spaces of D^s and D^t similar to each other allowing to get similar results for both domains. The most common domain adaptation methods rely heavily on either mathematical heuristics, which are formulated as loss functions affecting the latent space or a min-max problem formulated with adversarial learning.

Our method leverages adversarial learning only since mathematical heuristics can be added to any model to improve its performances. We assume that the classifiers suffer with the classification of new data since the latent space extracted contains several information about the input, which is not useful for classification. This information is due to the variation in the data style and is considered noise. Hence, we apply drastic transformations and data augmentation techniques to the input images and build a classifier that predicts the transformations applied to each image. This transformation-classifier is trained separately, and its loss does not influence the encoder part of our model (the feature extractor). On the contrary, the encoder is trained on the adversarial side of that classification loss, removing the transformation and style information from the extracted latent representations.

We evaluate our method on a new dataset of Medieval Musical Instruments annotated by five expert musical instruments historians and on toy datasets such as MNIST, SVHN, USPS. The images are extracted from three sources (AnnMusiconis, AnnVihuelas, MIMO), providing us with images of instruments through a long time period.

To conclude, in this chapter, we present two novel approaches to obtain domain adaptive model by introducing our separability loss, discrimination loss and classification loss which works by generating a latent representation that is both domain invariant and class informative by pushing samples from the same classes and different domains to share similar distributions. After examining the existing state-of-the-art contributions in DA, and comparing them against our model presented in this paper, we can conclude that our model surpasses them in its adaptivity, accuracy and complexity.

The rest of the chapter is organized as follows. Section [5.2](#) provides a related works presentation. Section [5.3](#) presents our initial model architecture TripNet and

our novel loss function. Section 5.4 presents our second model AugNet. Details of our experimental setup, and empirical results are shown in Section 5.5. Finally, Section 5.6 wraps up the chapter.

5.2 Related Work

In solving the problem of UDA, recent works used deep learning in various ways to build their models. Discriminator module was a core in most of the papers [67, 118, 126, 175] and its loss is used to tell if the features extracted from both domains are distinguishable or not. Within the works that have used the discrimination loss, several are based on generative models [67, 126] and its reconstruction loss [153, 78], and some have used pseudo-labeling [94, 67, 182, 169, 183, 139] to engage the target domain data into the process of classification. In this regard, our model is an example of the case where discrimination loss and pseudo-labeling are used without any reconstruction of the input images. We briefly touch upon these topics below.

5.2.1 Discriminator

The works [118, 126, 175] used the discriminator in the same manner. The discriminator was fed by the feature vectors of the source and target images and its loss is used to push the base/encoder/feature extractor network to produce indistinguishable features. Whereas in [67], the feature vectors were used to generate images in both source and target domains, and those images were fed into one of two discriminators that distinguished between real and fake images. This methodology was designed to ensure that the features extracted from the encoder network can be used to generate images of both domains; in other words, the features were domain invariant.

5.2.2 Image Reconstruction

Similar to image-to-image translation scenario, image reconstruction can be used in an Encoder-Decoder like architecture to drive the encoder to generate features for both domains that can reconstruct the image regardless of its domain. For example, in [78] and [153] a bi-shifting autoencoder (BAE) and an invertible (AE) used the reconstruction loss to convert samples between domains.

5.2.3 Pseudo Labeling

It is a commonly used method in semi-supervised learning. In UDA, pseudo labeling is used for narrowing the gap between the target and source domains by providing the pseudo labels for the unlabeled samples from the target domain. In [183], two classifiers were used to label the unlabeled target data, which was used further in training the rest of the components. On the other hand, in [139], the researchers used a similarity metric in building a K-NN graph for the unlabeled target samples and the labeled source samples. In DupGAN [67], the classifier built on the labeled source images was used to get the high-confidence images from the target domain, which were then used to train both the classifier and the discriminators.

5.3 TripNet: Category Based Adversarial Domain Adaptation

5.3.1 Overview

The following section describes the proposed model for UDA. We start by defining the notations that we used. The source domain images and labels are denoted as $X^s = (x_i^s, y_i^s)_{i=1}^N$ and the target domain images are $X^t = (x_i^t)_{i=1}^M$, both x_i^s and x_i^t share the same dimensions but different distributions. Since our research focuses on closed set domain adaptation, the target and source domain labels \mathcal{Y} are exactly the same. Our model contains an encoder, a classifier and a discriminator, as shown in Fig. 5.2. The encoder and the classifier make up the final classification model, and the discriminator is used to train the encoder to generate domain invariant features. Hence our classification function f is the composition of two functions $f = e \circ c$, where $e : \mathcal{X} \rightarrow \mathcal{Z}$ is the encoding function that maps the images into feature vectors and $c : \mathcal{Z} \rightarrow \mathcal{Y}$ categorizes the features for both domains. The discrimination function g is also the composition of two functions $g = e \circ d$ where e is the same encoding function and $d : \mathcal{Z} \rightarrow \mathcal{A}$ is the binary classification function that discriminates the domain of the latent representation of the input image.

In addition to the usual classification and discrimination loss, we introduced a separation loss that operates on the output of the encoder similarly to Linear Discriminant Analysis (LDA).

5.3.2 Architecture

Encoder: The encoder $E(\cdot)$ is a CNN like network with weights W^E . The target of the encoder is to produce the latent representation of both source and target

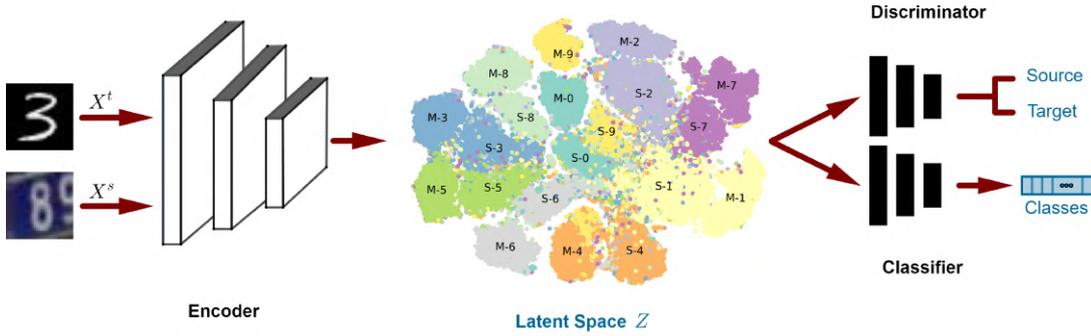


Figure 5.2: The Architecture of our model. It can be divided into three parts: an encoder, a discriminator and a classifier. The encoder translates the images (i.e., X space) to embeddings in the latent space (i.e., Z space). In the latent space, each group of embeddings is marked by either S_i or M_i , where i is the label of the image, and the prefix letter notates whether it is from MNIST (M) or SVHN (S). Thus, the Z space can be expressed as $Z = Z^s \cup Z^t$, where $Z^s = \bigcup_i S_i$ and $Z^t = \bigcup_i M_i$. The latent representation is fed to both the discriminator and the classifier. The discriminator distinguishes if the latent representation is from source or target domain, whereas the classifier finds the suitable label for it.

domain images as below:

$$z = E(x), x \in X^s \cup X^t \quad (5.1)$$

where $z \in Z$ are the extracted features that we aim to be domain invariant and category informative. Therefore, in case of input image from source domain x^s , the output of the encoder is $z^s = E(x^s)$ and if the image is from target domain x^t , the output is $z^t = E(x^t)$. The output of the encoder is fed to both the discriminator and the classifier.

Discriminator: Discrimination between the source and the target in the latent space is a core part in many of the recent contributions in DA as it was mentioned in Section II. Our Discriminator $D(\cdot)$ is a DNN with weights W^D . The discriminator is working as a binary classifier to label the latent representation of the images to one of the domains as follows:

$$a = D(z) = D(E(x)), a \in \mathcal{A}, \mathcal{A} = \{0, 1\} \quad (5.2)$$

Classifier: is a feed forward neural network $C(\cdot)$ with weights W^C for either binary or multi-class classification. It takes as input the latent representation z

and outputs the probabilities for each class \hat{y} . C can be any kind of DNN with a softmax output activation function that fulfills the tasks T_t and T_s .

In the case of multi-class classification, C predicts the class probabilities as follows:

$$\hat{y} = C(z) = C(E(x)), \quad x \in X, \quad X = X^s \cup X^t \quad (5.3)$$

where \hat{y} is the predicted class probabilities vector for the images of both domains $\hat{y} \in \hat{Y}$, $\hat{Y} = \hat{Y}^s \cup \hat{Y}^t$, and it shares the same dimensions as the one-hot-encoded source labels y^s and the target pseudo labels y^t . The classifier C and the encoder E are pre-trained on the source data alone and then used to generate the pseudo labels for the target domain Y^t [28, 136, 67], using the output of the classifier on the target images that C is highly confident about. In the beginning the number of samples chosen for pseudo labeling will be small or even zero and it increases as we get more domain invariant features.

5.3.3 Losses

Here, we introduce our three losses and explain how each one contributes to achieve domain invariant and category informative features.

Classification Loss: It is the usual cross entropy loss $H(.,.)$ for the output of source images and their labels and the output of target images (chosen for pseudo-labeling) and their corresponding pseudo labels, and is computed as below:

$$\mathcal{L}_c(W^E, W^C) = \left(\lambda_s \sum_{x^s \in X^s} H(\hat{y}^s, y^s) + \lambda_t \sum_{x^t \in X^t} H(\hat{y}^t, y^t) \right) \quad (5.4)$$

Where λ_s and λ_t are used to balance the weighted sum between the source and the target since we only take a few samples using pseudo labeling. By minimizing this loss, we update the weights of both the encoder and the classifier, W^E and W^C .

Discrimination Loss: In order to get domain independent features we used the discrimination loss to train the Discriminator to distinguish between the features for both domains using binary cross entropy as follows:

$$\mathcal{L}_D(W^D) = - \sum_{z^s \in Z^s} \log(D(z^s)) - \sum_{z^t \in Z^t} \log(1 - D(z^t)) \quad (5.5)$$

which can be written as:

$$\mathcal{L}_D(W^D) = - \sum_{x^s \in X^s} \log(D(E(x^s))) - \sum_{x^t \in X^t} \log(1 - D(E(x^t))) \quad (5.6)$$

where we assigned 1 for the source images and 0 for target images. The weights of the discriminator W^D are updated by minimizing this loss as an objective function. And the encoder is trained against an opposite loss that tries to put them in the same label as follows:

$$\mathcal{L}_P(W^E) = -\sum_{x^s \in X^s} \log(1 - D(E(x^s))) \quad (5.7)$$

\mathcal{L}_P is used to update the encoder so that the discriminator is deceived into thinking that the features extracted from both domains are indistinguishable. We also tried to deceive the discriminator using other loss variations, one aimed at pushing the domains in the exact opposite direction (i.e. target $\rightarrow 1$ and source $\rightarrow 0$) by adding $-\sum_{x^t \in X^t} \log(D(E(x^t)))$ to \mathcal{L}_P , and another loss aimed at pushing both the source and the target into the same label by adding $-\sum_{x^t \in X^t} \log(1 - D(E(x^t)))$ to \mathcal{L}_P in Equation 5.15, but both these variations didn't improve the results and made the model diverge with more iterations. \mathcal{L}_D and \mathcal{L}_P work against each other in a scenario similar to GAN discrimination loss and generative loss.

Separability Loss: Inspired by Linear Discriminant Analysis (LDA) to capture the separability, Ficher defined an optimization function to maximize the between-class variability and minimize the within-class variability. Using this idea, we defined the separability loss as follows:

$$\mathcal{L}_{sep}(W^E) = \left(\frac{\sum_{i \in Y} \sum_{z_{ij} \in Z_i} d(z_{ij}, \mu_i)}{\sum_{i \in Y} d(\mu_i, \mu)} \right) \times \lambda_{BF} \quad (5.8)$$

$$\lambda_{BF} = \frac{\min_i |Y_i^t|}{\max_i |Y_i^t|}$$

where Z_i is the set of latent variables that belongs to class i and it can be expressed as $Z_i = Z_i^s \cup Z_i^t$, which is the union of the sets of latent representation of both domains that have the same label i . Again, for the target domain latent representation, we used the pseudo-labels that are produced with a high level of confidence from the classifier. μ_i is the mean of the latent representations that has label i , so it can be expressed as $\mu_i = \text{mean}(Z_i)$, while μ is the mean of all the latent representations $\mu = \text{mean}(Z)$. $d(.,.)$ is a distance function we used to measure the dissimilarity between the latent vectors. So the numerator part of the equation is the the summation of the distance between each latent vector and its labeled-center, while the denominator is the summation of the distance from each labeled-center to the overall center of the latent representation. λ_{BF} is a balancing factor and is equal to the ratio between the number of least represented pseudo-labeled target samples $\min_i |Y_i^t|$ and the number of the most represented

ones $\max_i |Y_i^t|$, where i is the label. Its purpose is to pull the loss from converging to some local minima, if some classes are not well represented in the pseudo-labels. Thus, λ_{BF} keeps the separability loss from pushing the model into getting very high separability and accuracy in only a subset of the classes. Thus, by minimizing this loss function we can increase the separability of the latent representation according to the labels, regardless of its domain, which drives the encoder to lose domain specific features but preserve category informative features. It should be noted that the generator component in DupGAN [67] has the same purpose: it uses the extracted features to generate images in both domains to ensure that they are domain invariant, and reconstructs the input image to ensure that the features do not lose the category information. However, by using a loss function instead of a separate generator module, this work achieves the said purpose in a cost-effective manner.

5.3.4 Optimization

The overall objective function that we aim to minimize in this work is the weighted sum of the three losses, which we call the triplet loss, and is given as below:

$$\mathcal{L} = \min_{W_D, W_C, W_E} \beta_C \mathcal{L}_C + \beta_P \mathcal{L}_P + \beta_{Sep} \mathcal{L}_{Sep} \quad (5.9)$$

where $\beta_C, \beta_P, \beta_{Sep}$ are the balancing parameters. The detailed training process of our model, called TripNet, is described in algorithm 5.

Algorithm 4 : The training process of TripNet

Input X^s the source domain images, Y^s the source domain image labels, X^t the target domain images, $Epochs$ the number of epochs

Output Weights of the encoder W^E and the weights of the classifier W^C [1]

Pre-train E and C using X^s and Y^s **for** $e = 1$ to $Epochs$ **do**

end

Sample a batch of images for both domains $x^t, (x^s, y^s)$ Get pseudo-labelling y^t for x^t using C . Update W^D by deriving \mathcal{L}_D . Update W^C by deriving \mathcal{L}_C . Update W^E by deriving $\mathcal{L}_C, \mathcal{L}_P$ and \mathcal{L}_{Sep} . **return** W^C, W^E

5.4 AugNet: Augmentation Based Adversarial Domain Adaptation

5.4.1 Methodology

In this section, we describe our new method for unsupervised domain adaptation. Before we get into the details of our methods, we start by defining the notations used throughout the section. Let the source domain data be denoted as $X^s = (x_i^s)_{i=1}^N$ while the target domain data and annotations are $X^t = (x_i^t, y_i^t)_{i=1}^M$, it is important to note that the input dimensions of x_i^s and x_i^t are the same but they come from different marginal distributions. Since our research focuses on open set domain adaptation, the classes of the two domains overlap but not necessarily have to be the same.

Our model consists of an encoder (The main focus of our method), a classifier (used only for classification and is not influencing our method) and a transformation-discriminator (The added component of our method). The Encoder and the classifier are the final classification model as in a typical CNN classification scenario, whereas the transformation-discriminator is used only in training time and removed encoder at inference. Furthermore we can formulate our inference classification function f as the composition of two sub functions $f = e \circ c$, such that $e : \mathcal{X} \rightarrow \mathcal{Z}$ represents the encoder's function which performs the extraction of latent space vectors from the input images, and $c : \mathcal{Z} \rightarrow \mathcal{Y}$ performs the classification of the previously mentioned latent space vectors into their appropriate classes. The transformation-discriminator function g is similarly another composition of two sub-functions $g = e \circ d$ where e is the exact same encoder function whereas $d : \mathcal{Z} \rightarrow \mathcal{A}$ is the multi-class multi-label transformation detector function. Moreover, besides the commonly used classification loss and the discussed transformation-discriminator loss, we also used a separation loss which was shown to improve the results of domain adaptation models and it operates similarly to Linear Discriminant Analysis (LDA).

5.4.2 Architecture

In this subsection, we will only present the component of our method and in the next subsection we will see the losses that are used to train these neural networks.

Encoder: Our encoder $E(\cdot)$ is a typical pure Convolutional Neural Network with weights W^E (by default it contains only convolutional layers and max-pooling followed by a Flattening layer, but depending on the use of a pretrained model the architecture might include other layer types). The goal of using an encoder is

to encode the input images of both domain into a latent space representation in vector forms which is represented in the following formula :

$$z = E(x), x \in X^s \cup X^t \quad (5.10)$$

Such that $z \in Z$ represents the desired latent representation which we push towards being more domain invariant and category informative. Hence, we denote the output of the encoder for the source input images as $z^s = E(x^s)$ whereas for the target input images as $z^t = E(x^t)$. Both the classifier and the transformation-discriminator take as input the flattened output of the encoder.

Classifier: Our classifier is a vanilla artificial neural network (ANN) $C(\cdot)$ which is commonly used for multi-class classification or binary classification by changing the loss function between binary cross entropy and cross entropy (for the sake of our dataset we use the formulation with cross entropy as it is a multi-classification task). As previously states, its input is the output of the encoder function f which is the latent space vector representation z and its output is the per class the probabilities represented as the vector \hat{y} . The function we used in our case is the following:

$$\hat{y} = C(z) = C(E(x)), x \in X, X = X^s \cup X^t \quad (5.11)$$

In the above equation, \hat{y} represents the output of the classifier which the vector of per-class probabilities such that $\hat{y} \in \hat{Y}$, $\hat{Y} = \hat{Y}^s \cup \hat{Y}^t$ meaning it is common for both domains and for both labels of the target domain and pseudo labels for source domain. The first step is to train the classifier and the encoder on the target dataset only and we use the confidently predicted classes of source data as pseudo labels for later training. We repeat the pseudo labelling step on every iteration in the next step and it usually provides very few samples confidently in the beginning but it increases with time.

Transformation Discriminator: Our hypothesis is that the classifier isn't able to generalize well to other domains because the encoder is extracting information not just for classification but also about the style of the images. Our Transformation Discriminator $D(\cdot)$ is a Fully Connected Neural Network with weights W^D similar to the discriminator of the Generative Adversarial Networks but it has a multi-class output instead of a binary output. The transformation discriminator works in the following manner:

$$a = D(z) = D(E(x)), a \in \mathcal{A}, \mathcal{A} = \{[0, 1, \dots, 0], \dots\} \quad (5.12)$$

such that a is the predicted vector of the transformations applied to the image.

5.4.3 Losses

In this subsection, we give an overview of the three losses that we use to train our model.

Classification Loss: We start by explaining the classification loss as it is the most common loss in our method. As previously explained, we cross entropy loss $H(., .)$ applied on the predictions of target data and its annotations and source images and its pseudo-labels (if any), and is computed as below:

$$\mathcal{L}_c(W^E, W^C) = \left(1 * \sum_{x^s \in X^s} H(\hat{y}^s, y^s) + \lambda_t \sum_{x^t \in X^t} H(\hat{y}^t, y^t) \right) \quad (5.13)$$

Where λ_t is used to as a balancing hyper parameter between the two domains. As the loss function clearly states this loss effects both classifier and decoder in the same manner.

Transformation Discrimination Loss: The goal of the classification loss is to ensure the class informative quality in the latent space whereas the goal of the transformation discrimination loss is to ensure the domain independence quality of the latent space. order to get domain independent features we used the discrimination loss to train the Discriminator to distinguish between the features for both domains using categorical cross entropy CCE loss which operates on multi-label multi-class classification problems:

$$\mathcal{L}_D(W^D) = \sum_{x^s \in X^s} CCE(D(E(x^s)), Tr(x^s)) + \sum_{x^t \in X^t} CCE(D(E(x^t)), Tr(x^t)) \quad (5.14)$$

where $Tr(.)$ is the boolean vector of transformations applied to the input images. This loss effects only the weights of the transformation discriminator W^D .

On the other hand, the encoder is trained on the opposite loss that is maximizing the \mathcal{L}_D and trying to hide the information relative to transformation and style, as follows:

$$\mathcal{L}_P(W^E) = -\mathcal{L}_D \quad (5.15)$$

\mathcal{L}_P is the loss used to train the weights of our encoder component in order to deceive the transformation discriminator and remove the information relative to style and transformation.

Separability Loss: This is a typical example of a mathematical heuristic applied to an encoder as a method for improving the results and providing a cleaner latent space and hence an easier classification challenge for both domains. This loss is an extension of Linear Discriminant Analysis (LDA) which is in return an extension Fisher’s linear discriminant which is used to find a linear combination of features that characterizes or separates two or more classes of objects or events. It is later used a linear classifier to separate the classes. We use it as a continuous function trying to make the latent space as a combination of features that can separate the classes in the most linear way possible allowing the classifier to get a better generalization ability. It is defined as follows:

$$\mathcal{L}_{sep}(W^E) = \left(\frac{\sum_{i \in Y} \sum_{z_{ij} \in Z_i} d(z_{ij}, \mu_i)}{\sum_{i \in Y} d(\mu_i, \mu)} \right) \times \lambda_{BF} \quad (5.16)$$

$$\lambda_{BF} = \frac{\min_i |Y_i^t|}{\max_i |Y_i^t|}$$

Such that λ_{BF} is a balancing parameter used to reduce the effect of badly annotated source images.

Algorithm 5 : The training process of TripNet

Input : X^s — Source domain images
 X^t — Target domain images
 Y^t — Target domain image labels
 I — Number of iterations
Output : W^E — Weights of the encoder
 W^C — Weights of the classifier

Pre-train E and C using X^t and Y^t ;

for $i \leftarrow 1$ **to** I **do**

Sample a batch of images for both domains $x^s, (x^t, y^t)$;
Get pseudo-labelling \hat{y}^s for x^s using C ;
Update W^D by deriving \mathcal{L}_D ;
Update W^C by deriving \mathcal{L}_C ;
Update W^E by deriving $\mathcal{L}_C, \mathcal{L}_P$ and \mathcal{L}_{Sep} ;

end

return W^C, W^E

5.4.4 Optimization

To sum up, we can consider that our model is being trained to minimize the balanced loss which is a weighted sum of all the three mentioned losses, it is given in the equation below.

$$\mathcal{L} = \min_{W_D, W_C, W_E} 1 * \mathcal{L}_C + \beta_P \mathcal{L}_P + \beta_{Sep} \mathcal{L}_{Sep} \quad (5.17)$$

where β_P, β_{Sep} are the balancing parameters. We detail how our model improves its performance in Algorithm 5.

5.5 Experiments and Results

In this section we present the performances of our method on toy datasets commonly used in the field of unsupervised domain adaptation for the sake of comparison along with the presentation of the results on our annotated medieval manuscript studies datasets.

5.5.1 Toy Datasets and Results

We compare our model with several state of the art models on the famous SVHN-MNIST-USPS Benchmark to showcase the performances of our model. The models we will compare against are the current state of the art in the field of domain adaptation namely: DupGAN [67], SimNet [118], DANN [49, 48], ADDA [158], DSN [21], DRCN [54], CoGAN [100], UNIT [98], RevGrad [50], PixelDA [20], kNN-Ad [139] and ATDA [136] for digit classification. Since we followed the same experimental setup as most of the compared networks we will evaluate the models based on the accuracy on the target test set and compare it with the previously mentioned papers using their reported results from their original papers, then we will compare our model against DupGAN in terms of complexity (number of epochs).

We will also compare our model against itself, using just the Encoder and the classifier trained on the source domain only (noted as EC-SourceOnly) and the target domain only (noted as EC-TargetOnly) to have a lower bound and approximation of the upper bound. Most importantly we will showcase the importance of our method in the field of medieval manuscript studies and hopefully provide a good and solid empirical proof for the necessity and importance of using Domain Adaptation techniques in field where the data is scarce and limited.

Table 5.1: The test accuracy comparison for UDA on digit classification. The results for the previous works have been copied from the original papers or the DupGAN [67] without repeating the experiments because we used similar architecture for the encoder and the classifier part as well as the same experimental setup as those works. The "-" notation is used for experiments where the results have not been reported in previous works.

Paper	SVHN \rightarrow MNIST	MNIST \rightarrow USPS	USPS \rightarrow MNIST	SVHN _{extra} \rightarrow MNIST
ADDA	76.0	92.87	93.75	86.37
RevGrad	-	89.1	89.9	-
PixelDA	-	95.9	-	-
DSN	-	91.3	73.2	-
DANN	73.85	85.1	73.0	-
DRCN	81.97	91.8	73.0	-
KNN-Ad	78.8	-	-	-
ATDA	85.8	93.17	84.14	91.45
UNIT	-	95.97	93.58	90.53
CoGAN	-	95.65	93.15	-
SimNet	-	96.4	95.6	-
DupGAN	92.46	96.01	98.75	96.42
TripNet (Ours)	94.70	97.63	97.94	98.57

Digital Digit Recognition

We evaluated our model for unsupervised domain adaptation for digit classification task, on datasets with ten labels ranging from 0 \sim 9

MNIST database (Modified National Institute of Standards and Technology database) is the most commonly known machine learning database for handwritten digits recognition and is used for benchmarking almost every single image processing system. It contains a training set of 60,000 examples, and a test set of 10,000 example. It is a subset of a larger set available from NIST which was originally 20*20 images and converted into a 28*28 greyscale images centered around the center of mass of the pixels.

SVHN Street View House Numbers (SVHN) is created by taking pictures of real-world images used also for most benchmarks is association with MNIST. SVHN was created from numbers plates found in the Google Street View images and it provides a more challenging scenario than mnist because of the large amount of side artifacts in its images and since the images are RGB and not only greyscale.

USPS US Post Office Zip Code Data of Handwritten Digits which contains 7291 training samples and 2007 testing samples. The size of the images is are 16*16 grayscale but for the sake of our experiments we convert it into 28*28 make them similar to MNIST but overall less complex.

Table 5.2: The best set of hyperparameters for TripNet for the experiments reported in Table 5.1. β_{Sep} , β_P and β_C are the balancing parameters for the triplet loss from equation 6.6. λ_T and λ_S are the balancing parameters between the source and target classification losses from equation 7.1. PL_{Thresh} is the minimum confidence level provided by the classifier so that the image would be considered in pseudo labeling.

Experiments	β_{Sep}	β_C	β_P	λ_S	λ_T	PL_{Thresh}
SVHN \rightarrow MNIST	1.5	1	4	0.5	0.8	0.999
MNIST \rightarrow USPS	1.5	2.5	1	0.2	1	0.995
USPS \rightarrow MNIST	2.5	3	1.5	0.6	1	0.995
SVHN _{extra} \rightarrow MNIST	3	0.5	2	0.5	1	0.9999

Implementation Details

In all experiments, input images from all domains are reshaped into $32 \times 32 \times 3$ images, and each pixel was re-scaled to $[-1.0, 1.0]$. Given that the latent representation vectors z_i are high dimensional (512 in all experiments) we used the cosine similarity as our measure of distance $d(., .)$ in the separability loss in equation 7.4.

The Encoder part of our model has 4 convolutional layers using 5×5 filters with 64, 128, 256, 512 filters per each layer, respectively. The classifier and the discriminator are a 4 layer fully connected networks with 256, 128, 50 neurons per each of their first three layers, respectively, and an output layer with 10 neurons for the classifier and one neuron for the discriminator. The rest of the hyperparameters are reported below in Table 5.2 as they were tuned empirically for each experiment in Table 5.1.

Target Accuracy Comparison

Our model was evaluated for UDA for digit classification task, where the labels are $0 \sim 9$, using different datasets; MNIST of handwritten digits [88], SVHN of street houses numbers [113] and USPS [37]. These datasets were chosen because they have different distributions and their labels are present for validation and evaluation. We used 60000 images from MNIST from its training part and 10000 images from its evaluation part. USPS is a relatively smaller dataset from which we used 7291 images for training and 2007 images for testing. Finally, SVHN has 73257 images for training, 26032 images for testing and SVHN_{extra} has 531131 images for training, also. Our experiments were SVHN \rightarrow MNIST, USPS \leftrightarrow MNIST and SVHN_{extra} \rightarrow MNIST.

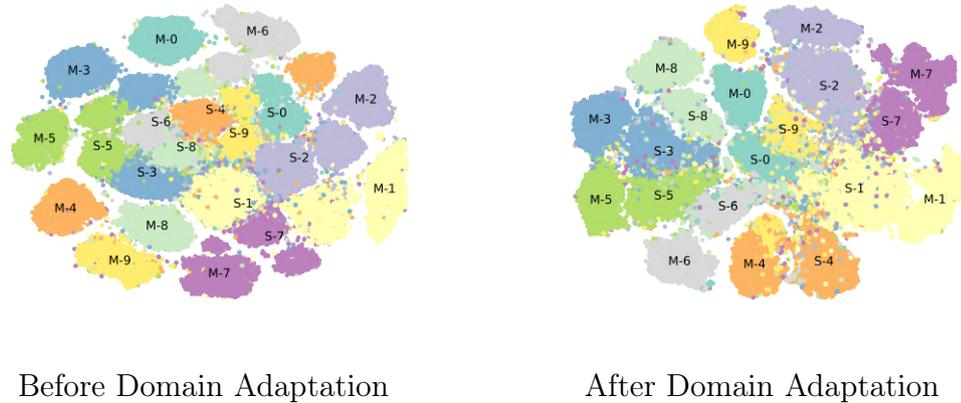


Figure 5.3: This is a projection of the latent representation of both source domain (SVHN) and target domain (MNIST), to describe their distribution in the latent space, before and after domain adaptation. Each cluster is labeled by M- i or S- i to denote if it belongs to the MNIST or SVHN, respectively, and i represents the label associated with each cluster.

The target accuracy results are shown in Table 5.1, where our novel model has either exceeded the compared methods or approached the highest achieved results. It is because of the use of the separation loss and the discriminator which allowed our latent representations to be domain invariant as seen in the Fig. 5.4 where all classes have been clustered together regardless of their domain.

Fig. 5.4 illustrates the projection of latent representation of both domains in the first experiment, SVHN \rightarrow MNIST. The projection for the visualization was produced using T-SNE [106]. In Fig. 5.4, it can be noticed that, after domain adaptation, the clusters for both domains that carry the same labels settled close to each other in the latent space, which demonstrates the competence of the presented model.

Comparison of TripNet and DupGan

We conducted a comparison between the convergence of TripNet and DupGAN for the experiment SVHN \rightarrow MNIST, as shown in Fig. 5.4. It's clear that our model converged significantly faster, after just first 120 epochs, than DupGAN which didn't even approach its max accuracy even after 500 epochs. Based on our experiments, we found that the generative model of DupGAN needs roughly 100 times the number of epochs TripNet needs to reach its maximum accuracy.

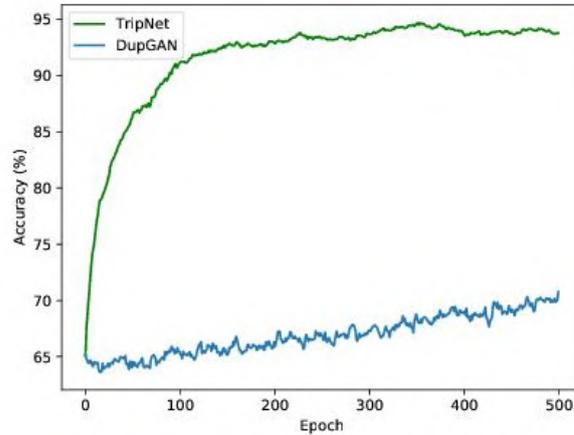


Figure 5.4: Comparison between TripNet and DupGAN in terms of number of epochs needed for convergence for SVHN \rightarrow MNIST case. This shows that the generative model comes with its high cost of training time.

Table 5.3: The test accuracy comparison for UDA on digit classification.

Target	Deep Source	DupGAN	TripNet	AugNet(ours)	AugNet- Sep (ours)
SVHN - MNIST	98.55	98.72	98.79	99.12	99.43
MNIST - USPS	95.02	96.70	96.24	98.23	98.90
USPS - MNIST	98.55	99.31	99.18	99.53	99.51
Avg	97.37	98.24	98.07	98.96	99.28

AugNet: Digital Digit Recognition

In order to evaluate our AugNet method, we implemented the following strategy. We used a dataset as a source without any labels and a target dataset with labels. We compared our method against the baseline of training only on the source dataset (Deep Source) and we compare our values against two state of the art methods in domain adaptation which are DupGan and TripNet. These two models are based also on adversarial losses but they use the same type of discriminator as GANs which takes a lot of time to stabilize whereas our method is much faster. We also compared the model with and without the separability loss to prove its efficiency.

We report the results in table [5.3](#) where we can see that all methods improve over the baseline on average which is expected and our method clearly outperforms the others on average and especially for the MNIST-USPS experiment. We also see the our method improves by using the separability loss on average. In these experiments we found that the following transformations gave the best results:

Table 5.4: Test accuracy for the ablation study for TripNet

Experiments	MNIST \rightarrow USPS	SVHN _{extra} \rightarrow MNIST
EC-SourceOnly	82.20	67.42
TripNet-WD	96.07	81.4
TripNet-WPL	90.42	71.06
TripNet-WSL	90.86	71.33
TripNet-WBF	96.71	92.83
TripNet (Ours)	97.63	98.57
EC-TargetOnly	97.36	99.12

Random Gray scale, Random Collor jitter (For the SVHN experiment), Random Scale, IAA Super pixels (SLIC algorithm), Bluring.

Ablation Study

For the purpose of seeing the usefulness of each component of TripNet, we performed an ablation study on MNIST \rightarrow USPS and SVHN_{extra} \rightarrow MNIST by running the experiments each time without a specific component and comparing it with our final model’s results, and the EC-SourceOnly and EC-TargetOnly. Our experiments also covered training without the balancing factor for separation loss (referred to as TripNet-WBF), training without the separation loss (referred to as TripNet-WSL), training without pseudo-labeling (referred to as TripNet-WPL) and training without discriminator (referred to as TripNet-WD). As shown in Table 5.4, the accuracy down-grades if we exclude any component which shows the necessity of each in the presented model to build an efficient architecture for domain adaptation. It is also worth noticing that our model (TripNet) got better results than EC-TargetOnly on the MNIST \rightarrow USPS even though EC-TargetOnly was trained directly on USPS data and that is mainly due to the fact that USPS and MNIST are very similar and USPS has a small training set.

5.5.2 Medieval Dataset Results

In this section we go through the dataset description and the results that we obtained on the different benchmarks. The datasets we used in this chapter are AnnMusiconis, AnnVihuelas and AMIMO which we have described in details in the dataset chapter.

Table 5.5: The test accuracy comparison for UDA on Musical Instruments Recognition In Medieval Artworks.

Target	Deep Source	TripNet	AugNet(ours)	AugNet- Sep (ours)
MIMO - AnnVihuelas	92.39	93.24	95.96	96.93
MIMO - AnnMusiconis	74.24	81.05	83.90	86.63
AnnMusiconis - AnnVihuelas	92.39	92.72	94.81	95.83
Avg	86.34	89.00	91.55	93.13

Musical Instruments Recognition In Medieval Artworks

For the purpose of future comparison with our method and new proposed benchmark, we implemented the following setup. We used the MIMO dataset as a source without any labels for two experiments (one for MIMO and one AnnVihuelas) and made an extra experiment where AnnMusiconis is the source and AnnVihuelas is target dataset. We compared our method against the baseline of training only on the source dataset (Deep Source) and we compared our values against the TripNet method. We present also the results of our model with and without the separability loss.

We report the results in table [5.5](#). Our method clearly again outperforms the TripNet model on average which in return also improves over the deep Source baseline as expected. We also see the our method improves by using the separability loss on average. In these experiments we found that the following transformations gave the best results: Random Gray scale, ISO Noise, Shift Scale Rotate, Random Collor jitter (For the SVHN experiment), Random Scale, IAA Super pixels (SLIC algorithm), Blurring.

5.6 Conclusion

The current models that approached the unsupervised domain adaptation problem using generative models are highly expensive to train in terms of time and space. Therefore, in this chapter, we made two main contributions which are two noval methods for UDA: TripNet, AugNet. TripNet consists of an encoder, a classifier and a discriminator. Both the classifier and the discriminator are stacked on the encoder. For each of the three components, we defined a specific loss: classification, discrimination and separability loss. These losses are used to train the components in a weighted manner. TripNet achieves the state-of-the-art performance on unsupervised domain adaptation for the digit classification task. As a further work in TripNet, we will evaluate our model on larger datasets and different tasks. Also, we will explore the problem of semi-supervised domain adaptation as we believe that TripNet will excel in it as well. AugNet provides a fast adversarial based

non generative technique to bridging domain gaps between dataset via performing style transformations and trying to force the encoder to forget information relative to the style allowing the classifier to improve its accuracy across several domains.

Our initial idea of defining a bad scenario that hurts the model and pushing Encoder to fight against it in an adversarial manner is quite innovative, but what if we have a lot more meta data about the our dataset than just a Boolean to define which source it is extracted from. This is quite common for datasets in the field of medieval studies. Most of the data comes from museums libraries where they annotate all the context of each manuscript. Examples can be information regarding the expected year of the manuscript redaction, the year it was found, the technique with which it was scanned, the kingdom or region it was redacted in, the type of manuscript, the style of the manuscript, the use of colors or black and white, the medium (eg Tempera colors, gold leaf, colored washes, pen and ink), the dimensions. At least a detailed textual description that describes the manuscript in general is always present. All of this data can and should be used to help us build better models and that's what we will evaluate over the next chapter when we combine domain adaptation with knowledge Graphs.

Chapter 6

Auxiliary learning: Knowledge Graphs & Domain Adaptation

The methods and some results presented in this chapter were published in:

V. Eyharabide, I. E. I Bekkouch, and N. D. Constantin. 2021. "Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition" *Computers* 10, no. 8: 94. <https://doi.org/10.3390/computers10080094> [41].

This chapter focuses on the use of knowledge graph embeddings as anchors for computer vision models allowing to building a more coherent latent space. The goal of the approach is to improve the performances of computer vision models on target datasets through the use of a larger source dataset and an accompanying Knowledge graph. This method is tested on our datasets AnnMusiconis, AnnVihuelas, and the knowledge graph is extracted from MusicKG.

6.1 Introduction

Although machine learning models and neural networks especially have improved drastically in the field of supervised learning, they still suffer from a big bias issue, bias towards the training data. Unlike humans who can learn more accurate abstractions on what makes an object an object (for example identifying a real dog in photos and an abstract painting of a dog as the same thing), neural networks aim at extracting visual clues that increase the probability of the presence of the object. This leads to neural networks being less reliable and hence less useful for extreme cases where changes in the forms, shapes, and external characteristics of the object change frequently, especially on inference data. This is precisely the case of cultural heritage data which already suffers from a huge within dataset variance but also from a larger between dataset variance.

Transfer learning (TL) emerged as a solution that tackles the problem of lack of data and resources for training a full neural network architecture from scratch. It started as a logical step after the ImageNet competition which encouraged large corporations to present new architectures of Convolutional Neural Networks. Currently, transfer learning is the default method to use for training any computer vision models especially mainly for three reasons:

1. Ability to reshape the input: unlike for tabular datasets where the content and number of variables differ from one dataset to the next, with images we can easily reshape the size of the input image directly using mathematical operations.
2. CNNs translation invariance: Since CNNs are based on the convolutional 2D operation which slides over the input image batch by batch using the same kernel, it is able to extract useful information from the image regardless of their position in the image allowing it to generalize to more cases than a traditional multi-layer perceptron [84].
3. CNNs hierarchic abilities: with the help of model interpretation methods, researchers were able to make the claim that CNNs like most neural networks are hierachical models but most importantly, CNNs extract lower level generic features (such as edges, corner, circles, etc) in the first layers of the model and tend to focus on more higher level and abstract non generalisable features (such as face, guitar, body, etc), which allows to reuse the first layers of a CNN on any dataset.

Cultural heritage data is challenging to acquire, demanding to label, and varies in style through different historical periods. First, finding medieval artwork images containing a particular type of musical instrument is difficult; the older the instrument, the fewer artworks that contain it are found. Second, since ancient artworks are generally damaged or deteriorated, experts may have difficulties in classifying the instruments. Finally, when dealing with images containing musical instruments from different historical periods, there are significant differences in how they were painted or sculpted. Besides, the instruments may differ according to the artwork supporting materials, such as paintings, manuscripts, photographs, or sculptures. Those difficulties make the training process on such heterogeneous images a more demanding task.

Knowledge graphs provide rich semantic context about the images' content that is useful to extract class-informative embeddings. This article's contribution is to add semantic information gathered in knowledge graphs when training neural networks with sparse and heterogeneous image datasets, which is the case of cultural heritage data. In this approach, we use knowledge graphs as an anchor

for our deep learning models to organize and direct the model’s focus and incorporate not only visual information in the training process but also global and more connected information. We evaluate our method on our collected image dataset of Medieval Musical Instruments, which was annotated and carefully verified by five musicologists specialized in medieval musical instruments. The images we use as source data and their knowledge graph are extracted from the AnnMusiconis dataset, whereas our target images came from the AnnVihuelas dataset.

The rest of the chapter sections are organized as follows: Section 6.2 is an overview of related works. Section 6.3 describes our model in detail. The empirical evaluation of our method is shown in Section 6.4. Finally, Section 6.5 summarizes the chapter.

6.2 Cultural heritage datasets

We evaluated our KG embedding-based domain adaptation approach on music iconographical data. The analysis of ancient artworks containing musical instruments brings valuable information on the instruments’ nature, physical characteristics, or playing methods. In this section, we present the image datasets and the knowledge graph used to test our proposal.

6.2.1 Medieval and Renaissance musical iconography as source and target domains

As we mentioned before, transfer learning aims to improve a model’s performance on a target domain by reusing a trained model on an already-known source domain. This article uses an image dataset of medieval musical instruments as the source domain and a renaissance musical instruments database as the target domain. Previously in [10], we presented a new manually annotated image dataset of historical musical instruments and a non-intrusive Transfer Learning method for object detection. While in [9], we proposed another method for unsupervised domain adaptation, which starts by applying style transformations to the input images and train a transformation discriminator module to predict these style changes. Based on these previous articles, we reused the lessons learned to detect chordophones in medieval artworks, to detect herein vihuelas (a Spanish renaissance chordophone) from a small collection of images. In this article, we combined knowledge graph embeddings with visual embeddings from the images and trained a neural network with the combined embeddings to take our methods a step forward. The two main datasets used are AnnMusiconis and AnnVihuelas which are presented in details in the datasets chapter.

Pythagoras playing the lute



 **Location (current)** : Ulm, Baden-Württemberg, Germany
Location type : Temple
Location (original) : Ulm, Baden-Württemberg, Germany
Century : 15
Dates : 1469 -1479
Technique : Sculpture
Material : Wood
Partner database : Musicastallis
Original title : Pythagore jouant du luth
Iconclass reference :
11Q71453 seats of the clergy, choir-stalls
48C7323 lute, and special forms of lute, e.g.: theorbo
48C70 'Musica', symbolic representations, allegories and emblems

Figure 6.1: An example of artwork in the AnnMusicIconis database



Figure 6.2: Four examples of Renaissance musical instruments in the Vihuelas database

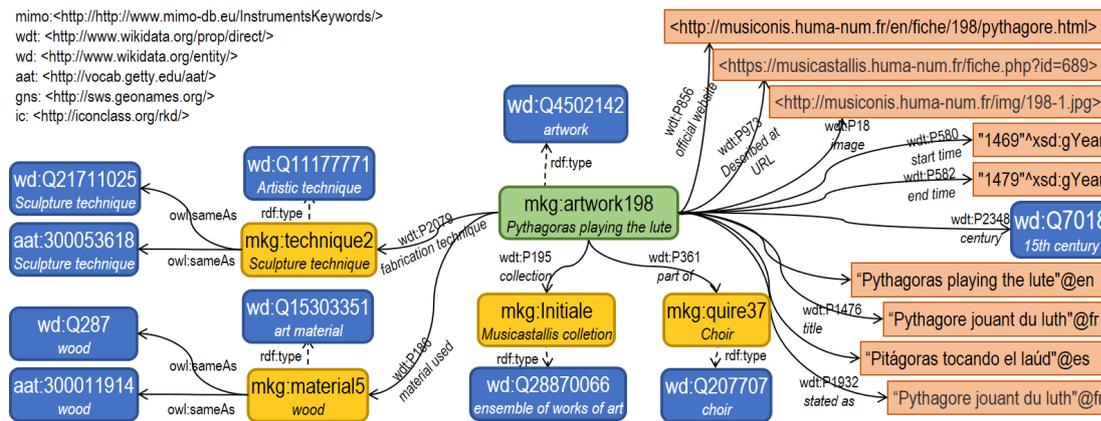


Figure 6.3: Representation of the artwork instance describing the example of Fig. (Eyharabide 2021) [6.1](#)

6.2.2 MusicKG: a knowledge graph of medieval musical iconography

In MusicKG, not only the artwork characteristics (such as artist, material used, or inception) are modeled, but also all the different scenes inside that artwork (for example, a couple dancing and a musician playing behind). In turn, each scene is described exhaustively by depicting the performer's characteristics (type, genre, clothing, position), the musical instrument's characteristics (type, family, material), the sound created, and the analogies, if any. Even though MusicKG is an extensive graph of relationships between performances and iconographic entities, we decided to use only a subset of all the RDF triples to create embeddings. Using a smaller graph allows us to better visualize and interpret the results. Once our approach efficiency has been proven, we plan to use all the available RDF triples to exploit the full potential of the KG.

The AnnMusiconis example presented before in Fig. [6.1](#)¹ [\[43, 41\]](#) (Eyharabide 2021) is depicted as a MusicKG artwork instance in Fig. [6.3](#). The main class is **Visual artwork** (herein "artwork") which is connected to the original sources through several predicates: **official website**, **collection**, **inventory number** and **described at URL**. Also, each artwork instance has a **title** from AnnMusiconis and a title from its original database in the **stated as** property. Generally, several **images** are associated with an artwork to capture all the details from different angles and different resolutions. Regarding dates, each artwork has **start time**, **end time**, and **time period** that indicate the century, the date on which the artist

¹https://www.mdpi.com/computers/computers-10-00094/article_deploy/html/images/computers-10-00094-g003.png accessed 3 September 2021;

could have begun and finished creating the artwork, respectively. The relation **material used** describes the material an artwork is made of, such as **Wood** or **Ivory** for sculptures; or **Textile** for embroideries and tapestry weavings. The relation **fabrication method** relates an artwork with its **Artistic technique**, such as **Sculpture** or **Painting**.

6.3 Methodology

This section describes our approach for domain adaptation using knowledge graph embeddings as anchors for our encoders. We start by presenting the terminology used, then detailing the components of our method, and finally describing the losses to train the different components.

The core of our idea came from the necessity of providing more data to the model in order to improve its results. Although hyper-parameter tuning techniques, Freebies, augmentations and regularisation techniques are quite useful, if you don't have enough data the model is just not going to train well. In our case, we are interested in museum data, which contains a lot of meta data about each manuscript. Although this meta data isn't necessarily related to the core of our musicology research and in most cases the annotators of these manuscripts completely ignore the musicology aspects of the manuscript, they still provide very useful information about the context of the image which can be used as input to the model to improve its results. The most common way to combine tabular (meta-data) and visual images is to use sensor fusion. By building a neural network that takes the meta-data and embeds it into a latent space, and does the same for the image, the latent spaces are then merged and fed into the same classifier. This allows the model to learn from multiple sources of input. the problem is although we have lots of meta data, they are very sparse and the model and can be used in situations where the meta data isn't available. Hence we decided to use the meta data in a way that it helps the model if it exists but keep the model architecture exactly the same allowing for an ease of application in multiple scenarios.

Before providing the mathematical functions and different losses, we establish the terminology and annotations used throughout the chapter. We denote the source domain as $X^s = (x_i^s, y_i^s)_{i=1}^N$ where x_i^s represent the input images with variant sizes, y_i^s is their respective classes, and N being the size of the source dataset. In this approach, it is important to note that the source domain is associated with a preexisting knowledge graph that connects the images of the dataset with concepts C^s in the graph, creating clusters of data around each concept. Thus, the images are linked together with not just the class information but also along several axes.

The target domain data is referred as $X^t = (x_i^t, y_i^t)_{i=1}^M$ where x_i^t represent the target images, y_i^t represents their respective classes, and M being the size of the

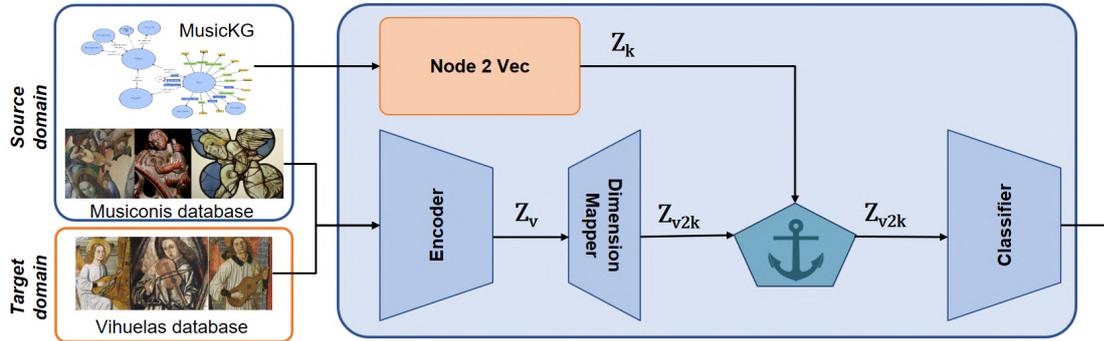


Figure 6.4: An overview of the proposed approach

target dataset. Domain Adaptation deals with the case where these two datasets share similar classes but usually come from a different distribution (meaning they have some large differences in terms of style, which lead neural networks to be unable to generalize). Since our research focuses on open-set domain adaptation, the classes of both domains overlap but do not necessarily have to be the same.

Our model contains two main components typically found in every computer vision classifier: the *Encoder* and the *Classifier* with an additional latent space mapper (the *DimensionMapper*) which converts the visual embeddings into the same dimensions as the knowledge graph embeddings (after training, it is considered to be a final layer of the Encoder). They are both present during the training and the inference phase. We can define our model function f as the composition of the Encoder function e and the classifier function c such that $f = e \circ c$ where $e : \mathcal{X} \rightarrow \mathcal{Z}$ maps the input images into a vectorial-1D latent space which is considered as the embedding of the visual information of the image, used later to classify it into its corresponding class using $c : \mathcal{Z} \rightarrow \mathcal{Y}$ which maps the embeddings space into the label space.

Our model's strength comes from a mathematical heuristic used to guide the neural network weight optimization through embedding anchors generated from the associated knowledge graph, which is based on Linear Discriminant Analysis (LDA).

6.3.1 Architecture

In this subsection, we detail the components of our method from a topology perspective. As shown in Fig. 7.3, the principal components are the *Encoder*, the *Classifier*, and *DimensionMapper*.

Encoder: Our feature extractor $E(\cdot)$ is a pre-trained pure Convolutional Neural Network extractor with weights W^E . In most cases it should contain only

elements such as convolutional layers and max-pooling with a Flattening layer in the end, but as different pre-trained model architectures exist, the components of our model might vary to include things such as residual blocks. The Encoder aims at transferring the shape of the input images from a 2D array with 3 channels to a 1D vector that fully connected neural networks can use, as shown in the following formula:

$$z_v = E(x), x \in X^s \cup X^t, z_v \in R^v \quad (6.1)$$

In our formulation, $z_v \in Z_v$ is the annotation for the 1D vector space extracted by the pre-trained Encoder, which represents (extracts) the visual embedding of the image. For differentiation between the visual embedding and the knowledge graph embedding, we use a lower case v to index the visual embeddings. The embedding space is a flat vector of the same size for all images. In this case, it is a set of v float numbers such that $v \in [256, 4096]$ depending on the model. The Encoder's output is sent to the space-mapper, which is later used for training the Classifier. The desired behavior for the visual embeddings is to be a class-informative, meaning it extracts information about the label of the object in the image, and domain-independent meaning images of the same class should be clustered together regardless of their original domain. In the rest of the section, we annotate the Encoder's output of source domain samples as $z^s = E(x^s)$ and the target samples latent space is $z^t = E(x^t)$. In the results section we use an encoder extracted from the pre-trained weights of Resnet 18 trained on ImageNet.

Dimension Mapper: we used a pre-trained model in order to improve the results of our models, this leads to having different latent spaces dimensions. The Dimension Mapper outputs the Encoder embeddings (which is in R^v) into the dimension of the knowledge graph embeddings (which is in R^k) as defined in the following operation:

$$z_{v2k} = DM(z_v) = DM(E(x)), x \in X^s \cup X^t \quad (6.2)$$

Such that our Dimension Mapper is a function that maps the values as follows $dm : \mathcal{R}^v \rightarrow \mathcal{R}^k$.

Classifier: We are using an Encoder/Classifier scenario. While the Encoder is the only pre-trained part and might include different layer types, the Classifier is a simplified fully-connected neural network (Multi-Layer Perceptron) $C(\cdot)$. This type of Classifier is commonly employed for all image classification cases, such as multi-class classification and binary classification, depending on the chosen final loss function (which might be a binary cross-entropy or cross-entropy). In this work, since our dataset contains several classes, we chose cross-entropy as the loss function. As indicated above, the input of our Classifier is the mapped visual latent

space, which was transformed from R^v to R^k where $k < v$ in our case. In contrast, the output is a 1-D probability vector that provides the model’s prediction and its probability of belonging to each class \hat{y} . The classifier function is defined as follows:

$$\hat{y} = C(z_{v2k}) = C(DM(E(x))), x \in X, X = X^s \cup X^t \quad (6.3)$$

In our annotations we use \hat{y} to reference vector of per-class probabilities, which is the output of the classifier such that $\hat{y} \in \hat{Y}$, $\hat{Y} = \hat{Y}^s \cup \hat{Y}^t$ meaning it is shared between both domains regardless. Before starting the domain adaptation process, we first train the Encoder, Dimension Mapper, and Classifier on the source domain until we reach convergence. Later, we use the Classifier’s output for the target domain as pseudo-labels, which will help us to better train the model, but only if the Classifier is confident in his decision. We set the threshold θ such that $\theta > 0.95$.

6.3.2 Losses

This section describes in detail the loss functions that our model uses to train its components .

Classification Loss: As expected, the first loss our model is trained on is the classification loss, which in our case is the cross-entropy loss function $H(., .)$ that reduces the differences between the probability distributions of the output and the labels since they are between 0 and 1 according to the following formula:

$$\mathcal{L}_c(W^E, W^C, W^{DM}) = \left(1 * \sum_{x^s \in X^s} H(\hat{y}^s, y^s) + \lambda_t \sum_{x^t \in X^t} H(\hat{y}^t, y^t) \right) \quad (6.4)$$

We use λ_t as a hyperparameter to control the contribution to the loss between the two domains. This loss affects the training of the Encoder, the Classifier, and the Dimension Mapper.

Anchoring Loss: The core contribution of our method relies on the anchoring loss, which takes the knowledge graph embeddings as anchors and brings the mapped visual embeddings closer to them, creating richer visual embeddings. This loss aims to embed more information in the Encoder’s training without using the data itself as input to the Classifier since it might be missing on some datasets and, most importantly, unavailable on new images. This loss allows our model to combine the increased accuracy of fusion models (which use multiple sources of input data, commonly images and text or images and a vector form tabular data) with the speed and generalizability of one source model used for image classification. This loss is based on a traditional machine learning model, namely, Linear Discriminant Analysis (LDA) and Fisher’s linear discriminant aiming to map input data into a space that linearly separates the samples.

Our goal is not to make the latent space linearly separable as this is impossible in the case of image embeddings and might lead to performance degradations. In fact, the objective is only to reduce the distance between the mean of the source latent space z_{v2k} which are linked to that specific concept allowing the latent space of the mapped visual embeddings to contain more rich information about the source images that the Encoder will be forced to focus on and extract. Our loss follows the following formula:

$$\mathcal{L}_{anc}(W^E, W^{DM}) = \left(\frac{\sum_{i \in Y} \sum_{c \in C} (d(\mu_{v2k}^c, a_k^c) + d(\mu_{v2k}^c, z_{v2k-i}^c))}{\sum_{ci \in C} \sum_{cj \in C} d(\mu_{v2k}^{ci}, \mu_{v2k}^{cj})} \right) \times \lambda_{BF} \quad (6.5)$$

$$\lambda_{BF} = \frac{\min_i |Y_i^t|}{\max_i |Y_i^t|}$$

The goal of this loss is: i) to reduce the distance between the center of mapped visual embeddings of a concept μ_{v2k}^c and their corresponding anchor a_k^c , which is represented in the loss as $(d(\mu_{v2k}^c, a_k^c))$, and ii) to reduce the distance between the center of mapped visual embeddings of a concept μ_{v2k}^c and its corresponding mapped visual embeddings z_{v2k-i}^c , which represented in the loss as $d(\mu_{v2k}^c, z_{v2k-i}^c)$. This formulation of the target proved to be faster in training than directly reducing the distance between the embeddings and their anchors one by one, and yet it provides the same gradients in the end. Our anchoring loss's second aim is to augment the distance between the centers of the mapped visual embeddings to create more space between them such that $ci \neq cj$. This loss is only used on different concepts and not the same one.

One downside of using this loss is that it usually has higher values than the classification loss and can influence the direction of classification. Besides, this loss is usually very imbalanced and depends on the random samples taken by the data loader and their classes/concepts, so we added a balancing factor λ_{BF} to reduce its effect when the classes/concepts are not balanced enough.

6.3.3 Optimization

We can imagine our model trained on the weighted sum of both losses. Considering the different ratios of loss weights (Classification Loss is usually 10-15 times smaller than the Anchoring Loss), the importance of the Classification Loss (which is vital for the classification), and the performances of our final model; we decided to add a balancing parameter β_A , which is usually around [0.01,0.03]. This optimization is summarized in the following formula:

Algorithm 6 : Knowledge Graph Embedding based Domain Adaptation

Input : X^s — Source domain images.
 Y^s — Source domain image labels.
 KG^s — Source domain knowledge graph.
 X^t — Target domain images.
 Y^t — Target domain image labels.
 β_A — Balancing factor - hyperparameter
Output : θ^E — Weights of the Encoder
 θ^{DM} — Weights of the domain Mapper
 θ^C — Weights of the classifier

```

// Creating the anchor embeddings  $a_k^c$  using node2vec
Sample walks using a random walk from the  $KG^s$ . ;
Embed the nodes of  $KG^s$  using the skip gram model. ;
Generate the  $a_k^c$  as the mean of the art work embeddings related to the
  concept;
// Pre-training The Encoder, Mapper, and classifier on the
  source domain.
for  $i \leftarrow 1$  to  $epochs$  do
  for  $j \leftarrow 1$  to  $nb\_batches$  do
    Sample a batch of source images  $(x_{1s}^j, y_{1s}^j), (x_{2s}^j, y_{2s}^j), \dots, (x_{N_s}^j, y_{N_s}^j)$ ;
     $\theta^E = \theta^E - \alpha \frac{\partial \mathcal{L}_C}{\partial \theta^E}$  Equation 7.1;
  end
end
// Anchoring the source visual concepts and adapting to the
  target
for  $i \leftarrow 1$  to  $I$  do
  Sample a batch of images for both domains  $(x^s, y^s, c^s), (x^t, y^t)$ ;
  Update  $W^E$  by deriving  $\mathcal{L}_C + \mathcal{L}_{anc}$ ;
  Update  $W^E$  by deriving  $\mathcal{L}_C + \mathcal{L}_{anc}$ ;
  Update  $W^C$  by deriving  $\mathcal{L}_C$ ;
end
return  $\theta^E, \theta^C$ 

```

$$\mathcal{L} = \min_{W_E, W_C, W_{DM}} 1 * \mathcal{L}_C + \beta_A \mathcal{L}_{anc} \quad (6.6)$$

For better understanding of the steps of our model, in Algorithm 7 we provide an algorithmic description that depicts the step-by-step operations needed for our method. The first step of our method starts by leveraging a knowledge graph that describes our source dataset, such as the century or the material used to create an artwork. We later create an embedding of each artwork based on its connections with the other nodes in the graph using the node2vec algorithm. These artwork level embeddings help us generate concept level embeddings which will be used as the anchors for training our neural networks. The second step is to train the neural network on Classification Loss, and minimize the overall distance between the center of visual concept embeddings and the normalized center of the knowledge graph concept embeddings. This method enables the Encoder part of the network to extract class-informative and structured latent space, allowing the Classifier to generalize better to other domains.

6.4 Results

This section evaluates our method’s ability to embed knowledge graph extracted information to improve the results of image classifiers on complex datasets that suffer from class imbalance and small sample sizes. We used two datasets: the An-nMusiconis dataset with its images, labels, and knowledge graph (MusicKG) and the Annvihuelas dataset with its images and labels. First, we describe the model’s abilities to generalize and present the per-class accuracy metrics against several baselines that do not use knowledge graphs and show that our model improves their results, proving the efficiency of adding knowledge graph data to computer vision deep learning-based models. Second, we evaluate our model’s performance when we change the source dataset’s size for training to show its sensitivity and resilience. Throughout the results section, all the reported results are the average of 5 runs of the model on the best hyperparameters found using k-fold cross-validation with k=10. The train-test split is a stratified split with 80% for training and 20% for testing.

6.4.1 Class level Evaluation

This subsection presents our enhanced performances against several baselines and shows that knowledge graphs can add value to computer vision models without altering the classification pipeline of classical deep learning-based image classification. We compare our model against three baselines: 1) *SourceOnly*: a deep

Table 6.1: Per class F1-score comparison between our model and three baselines.

Method	Source Only	Target Only	Source Target	KGE-DA (ours)	Metric
Viele	64.62	52.16	69.1	72.18	F1-score
	58.54	48.09	72.02	73.36	Precision
	72.1	56.98	66.4	71.03	Recall
Luth	53.92	67.14	74.96	85.63	F1-score
	50.1	62.96	73.22	82.25	Precision
	58.36	71.91	53.92	89.29	Recall
Bow	57.89	46.03	71.44	73.06	F1-score
	62.29	48.19	79.46	76.66	Precision
	54.06	44.05	64.88	69.77	Recall
Avg	58.81	55.11	71.83	76.96	F1-score
	56.8	60.55	75.8	74.37	Precision
	60.94	50.56	68.24	79.72	Recall

learning model sharing our architecture but trained only with the images and labels of the source dataset; 2) *TargetOnly*: a deep learning model sharing our architecture but trained only with the images and labels of the target dataset. 3) *SourceTarget*: a deep learning model sharing our architecture but trained with both the source and the target datasets’ images and labels.

We report the f1-scores for every main class in our dataset (Viele, Luth, Bow) and the macro F1-score for the models in table 6.1, since it strikes a good balance between precision and recall and evaluates the models much better than accuracy since the sample distribution amongst the classes differ broadly. We chose to use f1-scores for evaluation instead of accuracy as our dataset suffers from class imbalance and hence accuracy metrics are not very informative about the model’s performances. The table clearly shows that our method improves over the three baselines used for comparison. We can also see that the *SourceTarget* model outperforms both baselines since it uses the two datasets. Surprisingly the *SourceOnly* model outperformed the *TargetOnly* model on some classes even though it was not trained on the target data.

6.4.2 Target size Evaluation

In the previous subsection, we proved that our method outperforms the baselines. This subsection shows how our method performs against the *TargetOnly* and *SourceTarget* baselines when the target data’s size varies. This comparison is important since our method’s principal goal is to use datasets with tiny sample size (the general case of cultural heritage datasets) and still manage to get good results.

Table 6.2: Performance evaluation based on f1-score of KGE-DA method while varying target data sizes

Method	Source Only	Target Only	Source Target	KGE-DA (ours)
30%	58.81	36.14	60.31	67.03
45%	58.81	43.49	64.28	70.26
60%	58.81	49.26	64.42	74.86
75%	58.81	52.97	68.7	75.39
100%	58.81	55.11	71.83	76.96

As shown in Table [6.2](#), our model’s performances are always higher than the baselines, even in extreme cases. More importantly, our model’s performances were not affected as much as the baselines when reducing the target dataset’s sample size. We can also see that the *TargetOnly* baseline was the most affected even though it is the most used technique for small data cases. We can also see that the *SourceTarget* model still gives better performances than *SourceOnly* and *TargetOnly*. However, the drop of performance was significant, especially when going from 100% to 75% where it dropped from 71.83% to a 68.7%, unlike our model that only dropped 1.57% proving our method’s flexibility and efficiency even in extreme cases.

6.5 Conclusion

We presented a new approach to improve state-of-the-art domain adaptation methods using Knowledge graph embeddings. We combined knowledge graph embeddings with visual embeddings from the images and trained a neural network with the combined embeddings as anchors. This method is particularly appropriate when dealing with sparse and heterogeneous datasets, like those we generally face in the digital humanities and cultural heritage domain. We evaluated our approach on two cultural heritage datasets of images containing medieval and renaissance musical instruments. The experimental results showed a significant increase in the baselines and state-of-the-art performance compared with other domain adaptation methods. Besides, our model’s performances were not affected as much as the baselines when reducing the target dataset’s size.

Throughout the thesis, we made multiple assumptions about the data, which in most cases hold true. Yet, all these assumptions make the applicability of our research much harder, assumptions like the data always comes with large amounts of tags about the location, the artist etc. Such data although available for trusted sources such as big museums, it is not always available for all future datasets especially when it is manually collected by musicologists from different sources

such as social media or conferences and discussions. This led us to develop a new approach that makes the least amount of assumptions on the data, images are images, this is the only assumption we will make in the next chapter and we do that using a new domain generalization technique.

Chapter 7

Auxiliary learning: Domain Generalization

The methods and some results presented in this chapter were published in:

I. E. I. Bekkouch, D. C. Nicolae, A. Khan, S. M. A. Kazmi, A. M. Khattak and B. Ibragimov, "Adversarial Reconstruction Loss for Domain Generalization," in IEEE Access, vol. 9, pp. 42424-42437, 2021, doi: <https://doi.org/10.1109/ACCESS.2021.3066041> [8].

This chapter presents the idea of Adversarial Reconstruction Loss as a domain generalization technique that leverages the pixel values of the source dataset through an auxiliary task. The technique aims to push the encoder to forget pixel values, allowing the classifier to be more generic and less dependent on the style of the input. This method is tested on multiple toy datasets and our datasets AnnMusiconis, AnnVihuelas.

7.1 Introduction

As powerful as they are, Deep Convolutional Networks showed a huge dependency problem on the data set they were trained on, commonly known as over-fitting [61]. This problem (called domain-shift [156] or concept drift [148]) is mainly due to the fact that the training data set (Source domain) comes from a different distribution than the deployment data (target dataset), resulting in a decrease in the performance of the model [47], largely due to the fact that the latent distribution extracted by the encoders for both domain don't overlap, this can also be confirmed by using several Manifold Learning [104, 145] techniques as Bekkouch et al. showed [11] by reducing the dimensions of the rich latent space into a lower dimensionality and visualizing the distributions of both domains. Manifold Learning and domain generalization (deep learning in general) are both similar on many levels

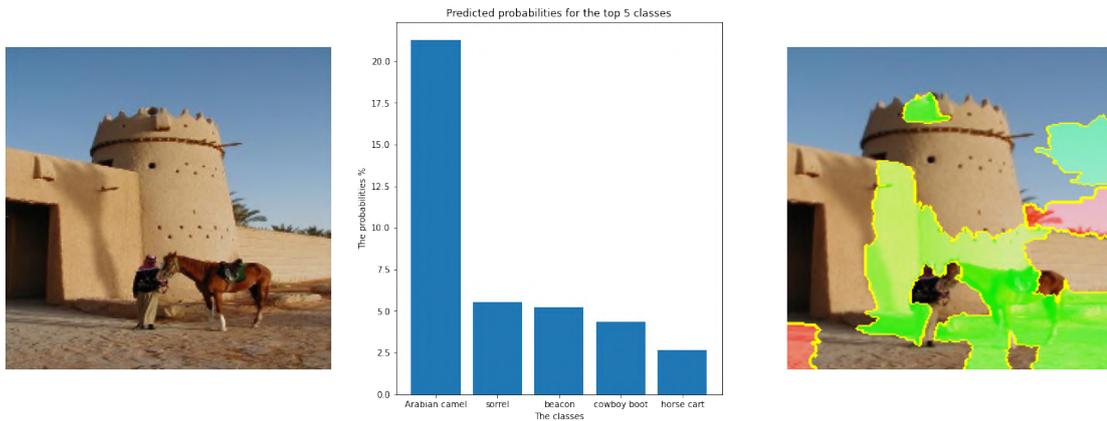


Figure 7.1: A horse wrongly predicted as an Arabian camel by ResNet, because of the surroundings. The left part is the LIME interpretation of the ResNet decision.

since they both reduce the input shape and learn an underlying structure in high dimensional data. The main difference between them is the ability for deep learning based feature extraction to include the class information in the latent space that is easily interpretable by a deep learning classifier unlike manifold learning methods which are mostly unsupervised or lack the easy integration with other deep learning components.

Such changes in real life can occur from very simple things like a change in image resolution or the brightness of the pictures or even changes in the background. As Fig 7.1 shows, the horse was miss-classified as an Arabian Camel by ResNet mostly because of the sand and Arabian architecture in the background, which the Local Interpretable Model-agnostic Explanations (LIME) [127] algorithm (used to interpret the decisions of black-box models per sample [128]) confirms by showing the pixels on which the ResNet relied on to make the decision. The same can be found in Fig 7.2 where a horse painting was misclassified as a macaw parrot because of the resemblance between their colors.

These big changes are at the heart of all of our challenges working in the medieval manuscript studies field and all historical and cultural preservation works. Our only chances for training a good model for such low level applications such as musical instrument recognition makes it really difficult to train a large model only on the medieval data. and in some cases we can't even train on such data because we haven't start the data collection task which is usually a highly time consuming and very low reward situation. Researchers in the cultural heritage field can really benefit from a method that doesn't require them to provide any low-level specific data for the objects they are looking for but only provide an example of this said object in the real world or even provide a set of hand-drawn sketches of the object.

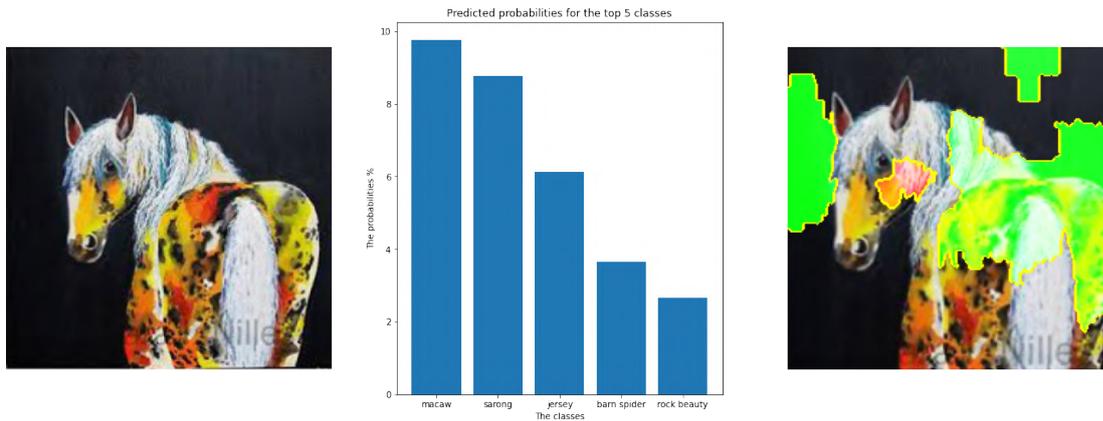


Figure 7.2: A horse wrongly predicted as a macaw parrot by ResNet, because of different colors (painting). The left part is the LIME interpretation of the ResNet decision.

The ability to make such a transfer is clearly very hard but such technique even with low quality results can speed up the search. Such problems are unavoidable in real research situations, which created a new field in transfer learning named Domain Generalization (DG).

DG can be also seen as a generalized case of the over-fitting problem, in the sense that the model is learning the data and not the task, even though in DG cases the model performs very well on the source test data, unlike traditional over-fitting scenarios. Domain Generalization (DG) [39] is a sub-field of Transfer Learning (TL) [116] that aims to solve the aforementioned problem by combining multiple data sources to train a more resilient model in hopes of generalizing to unseen domains. DG assumes the existence of multiple sources of data D_i^s (e.g. Photo, Art Paintings, and Cartoon) that are used for the same task T_i^s (e.g. classifying images of animals), and a target domain D^t (e.g. Sketches of the same classes of animals) that is harder to work with (harder to label or to collect). Most DG methods provide an extension to a closely related field, Domain Adaptation (DA) [7, 11] which often uses one source domain and one target domain to solve the domain shift problem. At the time of training, DA assumes the availability of target domain data but can be classified according to the presence of labels in the target domain in three key ways: Supervised [111], Unsupervised [11], and Semi-Supervised DA [36]. DG differs from DA in the fact that we do not have access to the target data nor its labels at training phase. Therefore, DG aims at building a model that can generalize well to unseen domains rather than generalizing to a single known domain.

Researchers have approached the problem of domain gaps and their conse-

quences in many ways. One traditional yet very commonly used technique is to treat this problem as an over-fitting problem and use regularisation techniques to help the model (parametric models) generalize well [149, 73]. Many techniques have proven to be useful in the case of deep neural networks such as learning rate decay, dropout [149], batch normalisation [73], L_1 , L_2 regularisation [114] and Shakeout [80]. Although these techniques were proven effective to help the model generalize well within the same data set and achieve higher test accuracy, however, it is not the most effective method for DG. Hence, we need to develop new methods that are both effective for over-fitting and for DG problems.

Recent approaches for DG are commonly neural-network-based and are separated into two main types: one-for-all and one-for-each. The former uses all source domains and learns a common model that works for all of them hoping it would generalize to future domains [23] whereas the latter approach (one-for-each), trains a different branch for each source domain. Next, at evaluation, we measure the closeness of each source domain to the target image and only consider the output of the corresponding classifier [92].

In this chapter, we deal with the case of one-for-all DG in its largest definition given its applicability and speed increase over the the one-for-each type. We implemented a new DG method that can generalize from multiple source domains to an unknown target domain, from one domain to another, and from one domain to itself, making this method easily applicable in many real world scenarios where the CNN or the neural networks in general show signs of over-fitting and dependency on the underlying distribution of the training data.

Similar to JiGen [23], who trains a jigsaw puzzle solver over the images to help the encoder better learn the internal structure, our approach belongs to the one-for-all category of DG approaches, focusing on how to use the training data more effectively to help the model learn better features in an unsupervised manner. In contrast to JiGen, the proposed model uses an Encoder, a Decoder, and a classifier to forget specific features of the data and not to learn it better. Unlike traditional Auto-Encoders that are trained to reconstruct the input, by training a Decoder to reconstruct the images and training the encoder in an adversarial way against the reconstruction loss, we force the Encoder to neglect the domain-specific details and only forward the information required for classification.

As proven by our experimental results on single source DG, our technique can also be helpful as a measure against over-fitting. Our approach uses pure deep learning based methods that can be run easily on GPUs, making it simpler to train and quicker to converge, unlike most other DG methods that add a huge computational burden such as JiGen (to make the jigsaw puzzle).

In short, this chapter presents a new DG system based adversarial auto encoders by training the encoder to extract only classification needed information

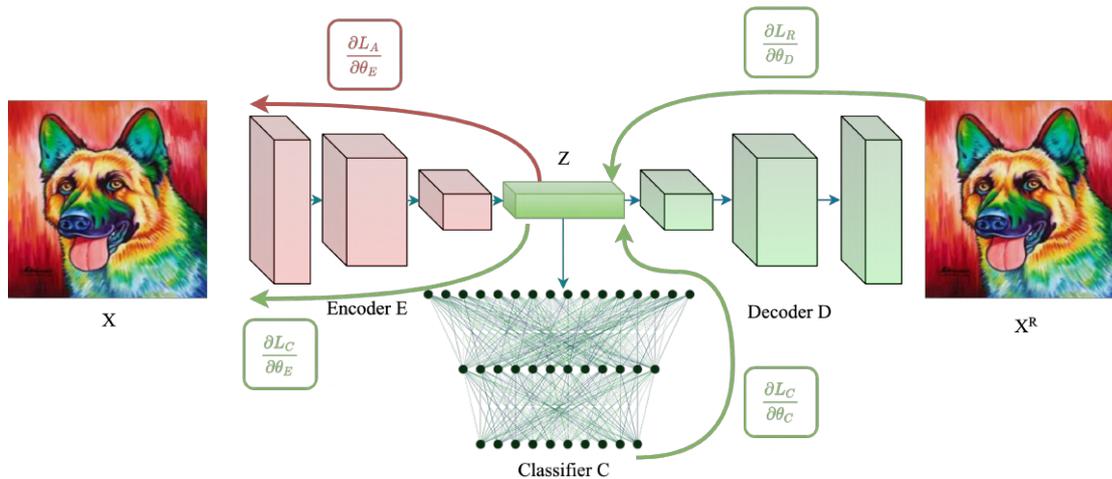


Figure 7.3: Model Architecture: The Encoder generates latent representation z which is used by the Decoder to reconstruct the input using L_R and by the Classifier to classify the sample using L_C . The encoder is trained on the classification L_C and adversarial L_A losses.

and remove all the style details noise, which achieves state-of-the-art efficiency in various scenarios for Domain Generalization, Domain Adaptation and Overfitting without adding a huge computational burden, making it more applicable to real-world scenarios and easily incorporated into more complex architectures. We evaluated our method against the state of the art deep learning methods based on five primary datasets and 13 sub-datasets and showed that our method outperforms most of them on all tasks.

7.2 Methods

We explain the approach of Adversarial Reconstruction Loss for Domain Generalization and the motivation behind it in this section. We base our approach on the premise that for the same problem, deep neural networks can not generalize to different domains because they are too dependent on their training domain. In other words, the CNN encoder portion is learning features that are helpful for prediction but also for extracting other domain-specific features that restrict the model's ability to handle unseen data. The CNN (Encoder) part of the models is responsible for the feature extraction; our main assumption is that the feature extractor extracts two types of information. Type 1 is the class-informative, which helps make the decisions and the classification, whereas type 2 is the misleading background noise. Thus, we characterize the model's ability for generalizing to

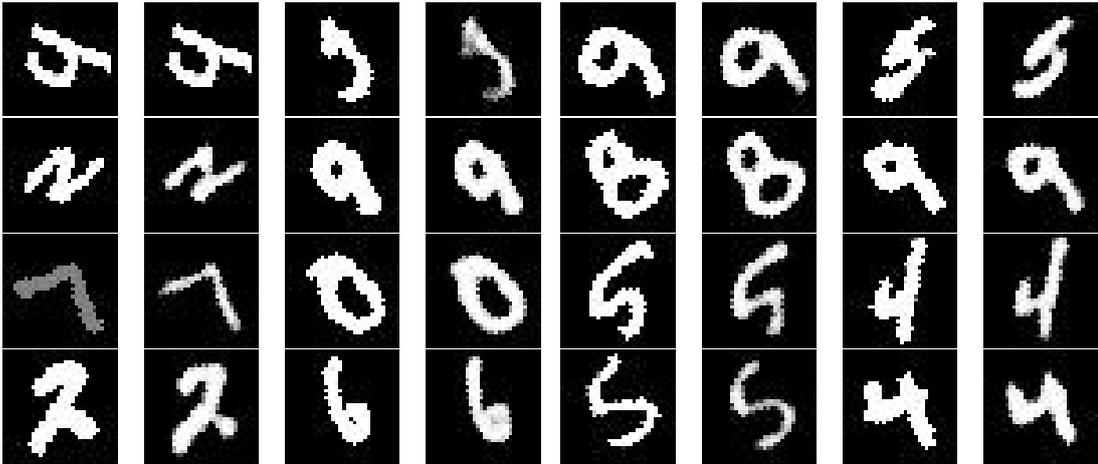


Figure 7.4: Reconstructed images formed by training a decoder on a model (Encoder+Classifier) trained only for classification. Reconstructed on the left, Input image on the right.

unseen datasets by its ability to forget the data’s peculiarities, symbolizing how much of the input has been overlooked or neglected by the encoder.

We illustrate the Encoder’s ability to sustain low-level image information despite the fact that the only loss we used for the training was the classification loss. Figure 7.4 explains the amount of information the Encoder preserves even after applying extreme input alterations.

After training an Encoder plus a Classifier setup on MNIST, the images were reconstructed based on a frozen Encoder and newly trained Decoder. These findings on the test dataset support our hypothesis that even though we train the encoder for classification only, it retains numerous input features from its source data.

7.2.1 Domain Generalization

As with all DG methods, our technique requires S source datasets (domains) and at least one target dataset (domain). N_i is used to represent the i th source dataset’s sample size, such that $X_i^s = \{(x_{i,j}^s, y_{i,j}^s)\}_{j=1}^{N_i}$, where $x_{i,j}^s$ references the j th sample of the i th source dataset and $y_{i,j}^s$ is its corresponding label. Moreover, we denote M as the target domain’s sample size with $X^t = \{(x_j^t, y_j^t)\}_{j=1}^M$, where x_j^t is the j th sample from the target dataset and y_j^t is its label, the t is used to distinguish between source and target domains.

The three main components of our model are: Encoder, Decoder, and a classifier, as shown in Figure 7.3. The central part of the model and our point of focus is the Encoder $E(\cdot)$ with its weights θ^E , which maps the input samples x into the

latent embedding space z . These features are commonly known as the images' latent representation.

The Classifier $C(\cdot)$ with weights θ^C , is a feed forward neural network and the whole classification model is the combination of the encoder and the classifier which is represented with the function $f_c = e \circ c$, where $e : \mathcal{X} \rightarrow \mathcal{Z}$ is the encoder function that maps the images into feature vectors and $c : \mathcal{Z} \rightarrow \mathcal{Y}$ is the classification function operating on the latent space.

The last part of our method is the Decoder $D(\cdot)$, which will not be included in the final model since it is not part of the inference process. Its weights are denoted as θ^D and we use it to reconstruct the input samples given their latent space representation such that the reconstruction function $f_d = e \circ d$ where $d : \mathcal{Z} \rightarrow \mathcal{X}$.

Each component of the architecture is trained with a different combination of losses, starting with the Classifier which is trained by minimizing the classification error (cross entropy loss) $H(\cdot, \cdot)$

$$\mathcal{L}_c(\theta^E, \theta^C) = \sum_{i=1}^S \left(\sum_{x_i^s \in X_i^s} H(C[E(x_i^s)], y_i^s) \right) \quad (7.1)$$

The decoder's weights are updated to reduce the reconstruction Loss (Mean Squared Error) between input sample x and the reconstructed image \hat{x} even though it doesn't have access to the input, it does that by mapping the latent space into a data sample.

$$\mathcal{L}_R(\theta^D) = \sum_{i=1}^S \left(\sum_{x_i^s \in X_i^s} \|D[E(x_i^s)] - x_i^s\|^2 \right) \quad (7.2)$$

Our method's crucial element is that the reconstruction loss \mathcal{L}_R will not be used to update the encoder's weights directly. Nevertheless, the encoder will be trained on both the classification loss and the adversarial of the reconstruction Loss:

$$\mathcal{L}_A(\theta^E) = - \sum_{i=1}^S \left(\sum_{x_i^s \in X_i^s} \|D[E(x_i^s)] - x_i^s\|^2 \right) \quad (7.3)$$

In computer vision, the initialization of the model's weights using an auto-encoder architecture and learning features useful for reconstructing the input is considered a standard best practice; and assumed to help build better classifiers using fewer data [95, 87, 44, 174]. We propose to take in the opposite route, enabling the Encoder to update its weights under the classification loss and skipping the structure, shape, and other information that overfits the network.

The step by step process of the training is described in Algorithm 7.

Algorithm 7 : Domain Generalization with Adversarial reconstruction

 loss

Input : X^s — Source domain images.
 Y^s — Source domain image labels.
generalizing_epochs — NB epochs 1
pretraining_epochs — NB epochs 2
 α — The learning rate
 β — Balancing factor - hyperparameter
Output : θ^E — Weights of the encoder
 θ^C — Weights of the classifier

```

// Start Pre-training the Model
for i ← 1 to generalizing_epochs do
  for j ← 1 to nb_batches do
    Sample a batch of source images  $(x_{1s}^j, y_{1s}^j), (x_{2s}^j, y_{2s}^j), \dots, (x_{N_s}^j, y_{N_s}^j)$ ;
     $\theta^E = \theta^E - \alpha \frac{\partial L_C}{\partial \theta^E}$  Equation 7.1;
  end
end

// Start the Generalization process
for i ← 1 to pretraining_epochs do
  for j ← 1 to nb_batches do
    Sample a batch of source images  $(x_{1s}^j, y_{1s}^j), (x_{2s}^j, y_{2s}^j), \dots, (x_{N_s}^j, y_{N_s}^j)$ ;
     $\theta^D = \theta^D - \alpha \frac{\partial L_R}{\partial \theta^D}$  Equation 7.2;
     $\theta^C = \theta^C - \alpha \frac{\partial L_C}{\partial \theta^C}$  Equation 7.1;
     $\theta^E = \theta^E - \alpha \frac{\partial (L_A + \beta L_C)}{\partial \theta^E}$  Equation 7.1, 7.3;
  end
end
return  $\theta^E, \theta^C$ 
  
```

Extension to Unsupervised Domain Adaptation

Our method is easily generalisable to the Unsupervised Domain Adaptation setting. Given the unsupervised nature of the Adversarial Reconstruction Loss, we can always add more samples without labeling which will help the model generalize even better. We also add in this setting a separation loss that operates on the output of the encoder similar to Linear Discriminant Analysis (LDA). The optimization goal is to maximize the between-class variability (making different classes

further apart from each other in the latent space) and minimize the within-class variability (making samples from the same class close together). Our separability loss is defined as follows:

$$\mathcal{L}_{sep}(\theta^E) = \left(\frac{\sum_{i \in Y} \sum_{z_{ij} \in Z_i} d(z_{ij}, \mu_i)}{\sum_{i \in Y} d(\mu_i, \mu)} \right) \times \lambda_{BF} \quad (7.4)$$

$$\lambda_{BF} = \frac{\min_i |Y_i^t|}{\max_i |Y_i^t|}$$

where Z_i is the set of all the latent representations of both source and target domains, that belongs to class i . For the target domain classes, we used the pseudo-labels that are produced with a high level of confidence from the classifier since we assume that the target data has no labels for training. μ_i is the mean of all latent representations with label i , such that $\mu_i = \text{mean}(Z_i)$, whereas μ is the mean of all the latent representations for both source and target $\mu = \text{mean}(Z)$. $d(.,.)$ is the distance function used to measure the dissimilarity between the latent vectors. λ_{BF} is a normalizer since the behavior of this loss is very fluctuating in cases where the batch doesn't contain a large enough amount for each class, and it represents the ratio between the number of least represented pseudo-labeled target samples $\min_i |Y_i^t|$ and the number of the most represented ones $\max_i |Y_i^t|$.

Extension to Over-fitting

Over-fitting arrives when a model has learned the training data too well. It is very common with strong models such as neural networks and decision trees. A number of techniques for combating over-fitting in neural networks exist such as reducing the model size, reducing the input data's dimensions, regularization (L1, L2), dropouts, and batch normalization, yet most of them constrain the model from actually learning category informative features.

Our technique although made for DG, can be easily applied in the case of single source datasets and contrarily to other over-fitting techniques, ours allows the model to learn as deep as possible without letting it over-fit on the style of the training data. Our method is not exclusive with other techniques, but it should be used along the side of most of the previously mentioned techniques since they are considered to be the best practice for the training process.

7.3 Analysis

Our Adversarial Reconstruction Loss method provided outstanding performances compared to other states of the art methods on several experiments using different

datasets. This section is split into four main parts; the first one is the Benchmarking datasets, where we present the five primary datasets and their 13 sub-datasets. The second part is the main results section, where we compare our model against several Domain Generalization methods on four benchmarks and especially for Medieval manuscript studies. The third and last parts are related to unsupervised domain adaptation and over-fitting results.

7.3.1 Benchmarking Datasets

To explore our Method’s effect on the domain generalization problem and its related issues (UDA, overfitting), we analyze five datasets extensively chosen in the field. The first one is **MNISTR**; the Rotated MNIST dataset is an alteration to the popular digits classification dataset MNIST. The different domains of RMNIST are created via rotating images by 15 degree increments: 0, 15, 30, 45, 60, and 75 (referred to as M_0, \dots, M_{75}). We employ a leave-one-out situation at the training phase, signifying that we will have five source domains and one remaining for the target. Nevertheless, the data has an identical test/train split as the primary MNIST; therefore, there is no overlap between train and test samples of the different domains. Next, we use the **MNIST-SVHN-USPS** Street View House Numbers (SVHN), a real-world image dataset for digit recognition which we used in our previous chapters. SVHN is obtained from house numbers in Google Street View images and is a little bit more challenging because of many side artifacts in it and the inclusion of color. US Post Office Zip Code Data (USPS) Handwritten Digits has 7291 train and 2007 test images. The images are 16*16 grayscale pixels which make them similar to MNIST but less complex. This combination of datasets is used both for Domain Generalization and Unsupervised Domain Adaptation.

7.3.2 Domain Generalization Results

digit classification: RMNIST

For the task of digit classification, we assessed our model’s performance versus numerous state of the art deep learning methods in domain generalization which are: MTAE [53], CAE [129], BSF [107], UDS [112], PSSO [163], AFLAC [3]. We were inspired to pursue this method after conducting experiments on the MNIST dataset to understand domain dependency better. Therefore, our model performs significantly better on this dataset than all the current state of the art, as Table 7.1 clearly shows our model’s performance exceeds all the other models on average and is ranked at least first or second in each experiment.

Table 7.1: Domain Generalization for digit classification: RMNIST. The average accuracy over 20 runs of the model. We represent each experiment by the name of its target dataset.

Method	0	15	30	45	60	75	mean
CAE [129]	72.1	95.3	92.6	81.5	92.7	79.3	85.5
MTAE [53]	82.5	96.3	93.4	78.6	94.2	80.5	87.5
PSSO [163]	94.2	82.5	96.3	93.4	78.6	80.5	87.5
UDS [112]	84.6	95.6	94.6	82.9	94.8	82.1	89.1
BSF [107]	85.6	95.0	95.6	95.5	95.9	84.3	92.0
AFLA [3]	89.3	98.8	98.3	93.3	97.4	88.1	94.2
ARL (ours)	89.5	97.2	97.3	98.1	96.7	89.4	94.7

The reported results are the averaged over 20 runs of the model with the learning rate set to 0.003, $generalizing_epochs = 50$, $pretraining_epochs = 100$, and the balancing factor set to $\beta = 0.1$. Our method outperformed all other methods on average providing more consistent results than others especially on the extreme case of 75 degrees, where we had 1.33% accuracy increase over the second best method AFLAC. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 5 hours and 46 min. The time needed to train the models for classification only without our loss is 2 hours and 18 mins.

In order to fully understand what our technique achieves we regenerated the experiment from Fig 7.4 but with adversarial reconstruction loss used for the training of the model. So our experiment goes as follows, We train the Encoder by the adversarial reconstruction loss and the classification loss as described in Algorithm 7 and after convergence, we re-train a new decoder on the latent space of the MNIST dataset without changing the encoder weights. After it converges, we evaluate the results on the test data with extreme rotations to see if the same effects from the previous experiment Fig 7.4 still holds. We inferred that the results in Fig 7.5 are definitely different in this case where most of the reconstructions appear to be centered and without rotation, unlike their respective original inputs. Furthermore, we can see that most of the specific details in the pictures tend not to appear in the reconstructed images. We can also easily see that all the reconstructions have the same class as their input. Proving that the aim of our method was actually achieved and that the learned features don't contain information about the specific details of the input yet they are still useful for classification.

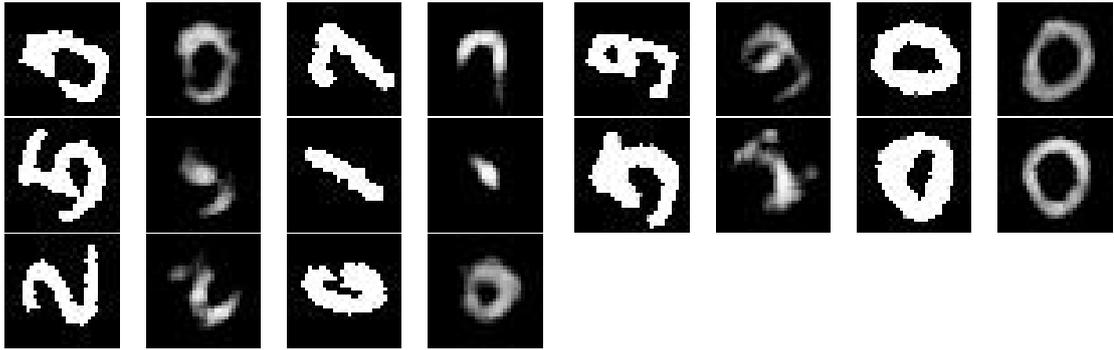


Figure 7.5: Reconstructed images formed after applying our ARL Generalization and training a new decoder to reconstruct the input images. Reconstructed on the right, Input image on the left.

Experiment		LB	DG	UB
Source	Target			
AMIMO+AnnMusiconis	AnnVihuelas	54.80	86.77	93.12
AMIMO+AnnVihuelas	AnnMusiconis	43.37	79.30	82.71
AVG		49.08	83.03	87.91

Table 7.2: The test accuracy comparison for DG on Musical Instruments Recognition In Medieval Artworks.

Musical Instruments Recognition In Medieval Artworks

Now for the main part of this chapter, we will take a look at whether this method for domain generalization is actually useful for Cultural Heritage applications and especially medieval musical manuscript studies. We used our three manually annotated datasets namely AMIMO, AnnMusiconis, and AnnViheuals. Training on AnnVihuelas and AnnMusiconis alone doesn't lead to improved results on AMIMO since the difference between the AnnMusiconis and AnnVihuelas is small and that doesn't help the model learn deeper features of the objects, and it makes it worse that the difference between AMIMO, and (AnnVihuelas and AnnMusiconis) is really big. Hence we will only experiment with AMIMO being a source dataset. We will compare our model's results against two benchmarks, the first UpperBaseline UB which refers to a model that is trained on AMIMO, AnnMusiconis, AnnVihuelas. and LB which refers to a model trained only on AMIMO and one extra dataset.

We reported the results of our musical instruments recognition in medieval artworks in the [7.2](#), before analyzing the results of our models we see that the results of the CNN trained only on AMIMO and one extra dataset leads to very

bad results on the unseen target domain unlike with domain adaptation where the target domain is at least present in the training, and this holds for both target domains (AnnMusiconis and AnnViheulas). Although our model’s results clearly aren’t comparable to the UB which is trained on all the datasets, our goal is not to surpass the UB but to get closer to it, which we clearly do especially when compared against the LB.

7.3.3 Unsupervised Domain Adaptation

In the case where the unlabeled target images exist during the training (Unsupervised Domain Adaptation), we add an extra loss to our model which is the Separability loss [7.4]. We explore the effects of this loss along with the performance of our model on two challenging scenarios, MNIST-USPS-SVHN dataset and the PACS data.

Digit Classification: MNIST-USPS-SVHN

This is the most common benchmark for domain adaptation tasks and UDA specifically. Hence we follow the same experimental setup as [11, 68]. We compare our results against first the two baselines (Upper Bound UB, and Lower Bound LB) which represent the accuracy of training and testing on the target dataset, and the accuracy of training on the source dataset only without access to the target dataset (not even unlabeled images), respectively. We also compare it against several of the state of the art deep learning methods in the field such as TripNet [11], DuplexGan [68], TarGan [105], Image2Image [99], Maximum Classifier Discrepancy [137], Generate to adapt [138], Joint Adaptation Networks [103] and Transferrable Prototypical Networks [117].

Table 7.3: Digit Recognition Benchmark on the MNSIT-USPS-SVHN dataset for Unsupervised Domain Adaptation. Each experiment name follows source_domain - target_domain naming convention. ARL-sep is used to reference to our method + the seperability loss and ARL is used to reference our model without it. The “-“ notation is used for experiments where the results have not been reported in previous works.

Method	UB	LB	JAN [103]	Gen2Adpt [138]	MCD [137]	I2I [99]	TarGAN [105]	DupGAN [68]	TPN [117]	TripNet [11]	ARL-sep	ARL
SVHN - MNIST	98.97	62.19	78.4	92.4	93.6	90.1	98.1	92.46	93.0	94.70	98.7	93.81
MNIST - USPS	95.02	86.75	84.4	92.8	90.0	98.8	93.8	96.01	92.1	97.63	98.3	97.12
USPS - MNIST	98.96	75.52	83.4	90.8	88.5	97.6	94.1	98.75	94.1	97.94	97.14	95.31
SVHN _E - MNIST	98.97	73.67	-	-	-	-	-	96.42	-	98.57	98.76	97.13

Our learning rate is $\alpha = 0.01$, *generalizing_epochs* = 250, *pretraining_epochs* = 200, and the balancing factor is set to $\beta = 0.15$. Table [7.3] shows that our method

Table 7.4: Multi-source Unsupervised Domain Adaptation results on PACS datasets obtained as average over five runs for each experiment.

PACS-DA	photo	art paint.	cartoon	sketches	Avg.
ResNet 18 [65]	92.9	74.7	72.4	60.1	75.0
Dial [24]	97.0	87.3	85.5	66.8	84.2
DDiscovery [108]	97.0	87.7	86.9	69.6	85.3
JiGen [23]	97.9	84.8	81.1	79.1	85.7
ARL-sep	98.3	86.1	87.6	73.4	86.3
ARL	96.5	82.9	83.9	71.7	83.7

outperforms most of the current state of the art techniques in 2 out of 4 experiments and ranked 2nd in the other two being only a few 0.05% away in the MNIST-USPS experiment. We can also see that our ARL-sep model outperforms our ARL model on all experiments, demonstrating the efficiency of the separability loss, yet it is also worth mentioning that the ARL model alone performed nicely being only 1.18% behind ARL-sep in the MNIST - USPS. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 1 hours and 32 min. The time needed to train the models for classification only without our loss is 0 hours and 31 mins.

PACS - Multi-source Domain Adaptation

Multi-source Domain Adaptation is a subset of DA where we have multiple source domains with labels but they are treated as one source, and a target domain either with or without labels. We are focused on the unsupervised case where the target domain is only available with images. Our method is unsupervised at its core making it easily applied in such case. To verify our assumptions we make the same experimental setup as other deep learning methods such as JiGen [23], DDiscovery [108], and Dial [24] by using ResNet18 [65] as our base model (Encoder + Classifier), whereas our Decoder is built as the mirror of the Encoder. We compare our method against all of the previous models and against a ResNet18 only model as our lower baseline. Our learning rate is $\alpha = 0.003$, *generalizing_epochs* = 350, *pretraining_epochs* = 500, and the balancing factor is set to $\beta = 0.1$. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 12 hours. The time needed to train the models for classification only without our loss is 8 hours and 12 mins.

The results in Table 7.4 summarize the outcome of this experiment, where the provided accuracies show that our method ARL-sep is superior to the other

Table 7.5: Accuracy results of different models on digit classification datasets MNIST-USPS-SVHN and MNISTR for the Over-fitting scenario. The best model is bolded and the second best is underlined.

Method	OF	WT	T-ARL	O-ARL	F-ARL-sep
MNIST	63.74	98.97	<i>99.31</i>	94.73	99.54
USPS	72.41	95.02	98.12	96.41	<i>97.93</i>
SVHN	58.46	94.97	<i>97.85</i>	92.9	98.14
Avg.	64.87	96.32	<i>98.42</i>	94.68	98.53
MNISTR					
0	63.74	98.97	<i>99.31</i>	94.73	99.54
15	60.13	96.64	98.07	91.93	<i>97.17</i>
30	68.52	98.03	<i>98.69</i>	92.86	99.05
45	68.24	98.14	<i>98.83</i>	94.33	99.29
60	65.05	97.12	<i>97.41</i>	92.74	98.17
75	62.48	<i>97.59</i>	97.43	93.42	97.84
Avg.	64.69	97.748	<i>98.29</i>	93.33	98.51

techniques on average and on two out of four of the experiments which are Photo target domain and the more difficult task of Cartoon target domain. We can also see that even though the ARL only model isn't outperforming the other methods but it still way better than the baseline with a 8.78% increase in accuracy on average and a maximum of 11.64% accuracy increase on the Sketches dataset.

7.3.4 Over-fitting

Over-fitting problems have been explored ever since the start of neural networks. Given the strong ability of neural nets to remember and memorize data samples. To evaluate the efficiency of our method on this problem we make the following setting, Train a model longer than it needs to force it to over fit, and then see if adding our loss can help bring it back from the over-fitting scenario, we refer to this model as (O-ARL).

We compare our method against several baselines: (i) Over-fitted model (OF), (ii) Well trained model (WT), (iii) model trained with ARL only from the start (T-ARL), and (iv) model fine-tuned with ARL-sep (F-ARL-sep). We perform this experiment on several benchmarks for digit classification which are: MNIST,

Table 7.6: Hyper-parameters for the over fitting experiments on digit classification Table 7.5. G-epochs is generalizing epochs and PT-epochs is pretraining-epochs.

Hyper-paramter	α	G-epochs	PT-epochs	β
<i>MNIST</i>	<i>0.01</i>	<i>50</i>	100	<i>0.2</i>
<i>USPS</i>	<i>0.01</i>	<i>50</i>	<i>100</i>	<i>0.15</i>
<i>SVHN</i>	<i>0.003</i>	<i>250</i>	500	<i>0.15</i>
<i>MNISTR</i>				
<i>0</i>	<i>0.01</i>	<i>50</i>	100	<i>0.2</i>
<i>15</i>	<i>0.007</i>	<i>100</i>	<i>200</i>	<i>0.25</i>
<i>30</i>	<i>0.007</i>	<i>100</i>	250	<i>0.15</i>
<i>45</i>	<i>0.003</i>	<i>250</i>	500	<i>0.1</i>
<i>60</i>	<i>0.003</i>	<i>250</i>	500	<i>0.15</i>
<i>75</i>	<i>0.003</i>	250	<i>500</i>	<i>0.1</i>

SVHN, USPS, MNISTR-0, ... , MNISTR75. For each one of these experiments we used a different set of Hyper-parameters which are all mentioned in Table 7.6. We use the same experimental setup as [140]. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 14 hours and 52 min. The time needed to train the models for classification only without our loss is 2 hours and 12 mins.

Table 7.5 shows the results of our over fitting experiments. The most obvious conclusion we can make is that the F-ARL-sep model, which was first trained on the data and then fine tuned with both the Adversarial Reconstruction Loss and the Separability loss, outperforms all the other models in most cases specifically the models that suffer from over-fitting OF and those who are well trained WT proving that our method is quite good for increasing model’s performances and accuracy even on the same data domain. We can also see that O-ARL model which was used on top of an over-fitted OF model was able to help the model go back to performing good even though it was not as good as F-ARL-sep but it still gave an increase of 29.81% in accuracy on average. We also see that the T-ARL model which is trained from the beginning on the ARL loss was as rigid as O-ARL and even better than WT model in most of the cases.

We also confirm our findings through Figure 7.6 where we show the behaviour of our different losses and how they influence the testing accuracy of the model on the MNIST dataset. We can easily notice that the over-fitted models always go up and then quickly decreases in performance as shown with OF chart, which

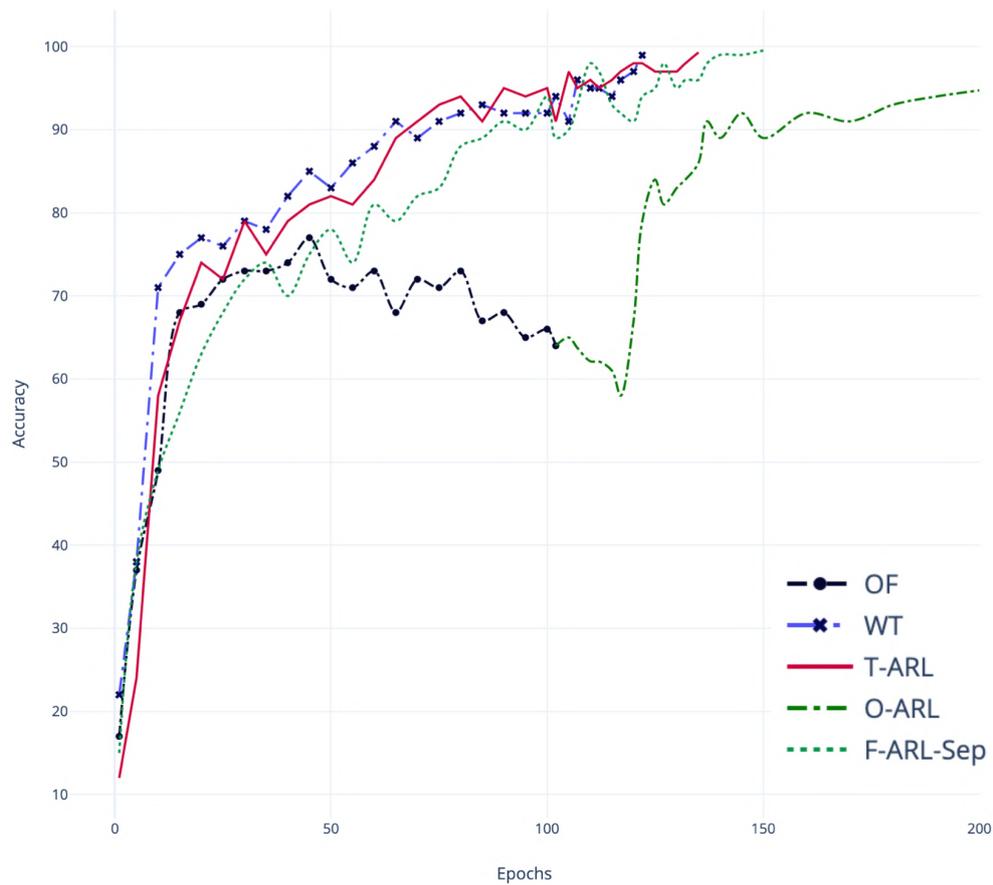


Figure 7.6: Comparison of different models on the task of digit classification on MNIST for the over-fitting scenario. The accuracy results are reported as the average of 5 experiments with the best hyper-parameters. OF is the over-fitted model, which is used by O-ARL as the initial start for solving the over-fitting problem. WT is the well trained model, T-ARL is the model which is trained from the start with ARL, and F-ARL-Sep is the WT model and fine-tuned with both ARL and sep loss [7.4](#).

is continued using the O-ARL chart which drops the performance in the first few epochs but then quickly starts giving positive outcome on the model's performance approaching results provided by the WT models. We can also notice that the WT models achieve better than our models in the first few epochs where as our models (F-ARL-Sep and T-ARL) improve slower but with enough epochs they exceed the WT performances.

7.4 Conclusion

We proposed a simple but effective task agnostic method for Domain Generalization and Unsupervised Domain Adaptation that is based on the assumption that models extract two types of information, class informative -useful- and style information -harmful-. Our method pushes the model to forget the style information while keeping the class informative part of the input which leads to high performance increase on several Object detection and classification benchmarks for DG and UDA. Our method also showed a great effect in fixing over-fitted models as shown by the experimental results. Moreover, the proposed method shows great promise of wide applicability since it is implemented orthogonally to other models and hence can be applied to different problems such as facial recognition without having to change the underlying algorithms.

Chapter 8

Musiconis: Populating the database while advancing research

Medieval musical iconography is an essential tool for anyone wishing to understand medieval music. This visual evidence is even more important when it comes to studying societies that remained faithful to the oral traditions of transmitting music or learning from instruments that are long gone and forgotten because of wars and time. Our only remaining portal to such periods are the texts and images that remain in the historical manuscripts at big Museums such as The Bibliothèque nationale de France (BNF) and Gallica which holds over 45000 different documents from the medieval period.

Medieval manuscripts are incredibly useful for musical studies because they provide a unique glimpse into the history of music. These documents contain invaluable information about the development of musical notation, as well as detailed descriptions of different musical styles and techniques. By studying these manuscripts, researchers and musicians can gain a deeper understanding of the cultural context in which certain pieces of music were created, as well as the techniques and methods used by medieval composers and performers. Additionally, medieval manuscripts often include beautiful illustrations and artwork that offer a window into the visual culture of the time period. Overall, medieval manuscripts are an essential resource for anyone interested in studying and performing medieval music.

Throughout this thesis, we collaborated very closely with multiple musicologists and other PhD students working on reconstructing an analysis of medieval vocal practices and medieval instruments. All such examples require musicologists to search through pages of manuscripts in hopes of finding a dozen images that allow them to validate their hypotheses.

The goal of musiconis is to build a representation of sound and music in the medieval ages. Musiconis allows researchers interested in medieval music to search in a

much more precise database of musical performances (musicians, singers, dancers) featured on medieval media (VIIIth-XVIth centuries). Musiconis is hosted online and allows users to search in over 2700 well-documented images in French, English, and Spanish. This great effort is the fruit of a big collaboration between musicologists under the supervision of Professor Frederic Billiet who were able to retrieve these manuscripts from multiple databases.

This chapter of our thesis is dedicated to showing the impact of artificial intelligence on medieval manuscript studies in a practical way. We will start by discussing two of the musicology use cases we collaborated on, mainly, Medieval singing understanding and musical instruments search. We follow that by showing how we used our algorithms to search for visual indicators that the musicologists give us as clues for what they want. We later discuss the positive results that we got, such as the big increase in new manuscripts added to musiconis, but also the failures of the model on specific use cases. We conclude this chapter by showing the positive part of AI work in general and especially in low researched fields which is the continuous improvement principle and how can we leverage it in an iterative approach to make our models better and advance research in musicology.

8.1 Project: Medieval Singing

The analysis of medieval vocal practices is an essential issue for musicologists and performers [15]. However, since the human vocal cords cannot be displayed explicitly, it is hard to identify whether a person or group is singing or not.

Even though there are numerous descriptions of musicians and singers in medieval chronicles and tales, illuminated manuscripts are the principal source for musical iconography. Illuminations often depict very complex situations in a tiny space. Artists often wished to concentrate much more information in a small illumination than would be contained within that scene in real life. However, studying a large corpus of images allows musicologists to detect repeated patterns and shed light on previously unknown medieval vocal practices across different periods and regions. The discovered patterns could enable performers wishing to perform repertoires to better understand the organization of singers, the environment, and the setting of the songs according to the period and genre considered. For example, considering the architectural modifications over the centuries, better choices regarding locations and musicians could be made to recreate acoustics as close as possible to the original music scene. Therefore, our objective is to find singing performances in images from medieval artworks. More precisely, we will detect medieval images containing persons in solo or group-singing situations, whether accompanied or not by musical instruments. The final objective for musicologists is to better understand the physical postures of singers, their relationship, and their

location inside the building.

Since the human voice is not a visible musical instrument, it is necessary to define possible objects in the images that may suggest the presence of singing performances. Therefore, we propose identifying characters who have their mouths open, perhaps with features linked to the vocal utterance (such as declamation or singing; see Figure 4.1a (BnF ms. fr. 166 f. 121v: <https://gallica.bnf.fr/iiif/ark:/12148/btv1b105325870/f256/1500,750,1000,1300/full/0/native.jpg>, accessed on 1 November 2021)). However, having the mouth open is not a sufficient condition to determine that a person is singing. The context or environment in which these singers are performing is vital to understanding the musical scene.

8.2 Project: Medieval Musical Instruments

Medieval Musical Instruments clearly indicate musical representations in medieval manuscripts, much easier than indices such as the open mouth or hand positions. This is why Musiconis is highly interested in collecting as many folios, manuscripts, stained glass, carvings (wood, ivory), and sculptures. The goal is to speed up the hypothesis-testing phase of musicology research. Medieval musical instruments are an important source of information for musicologists, as they can provide insights into the development of music in the Middle Ages. Manuscripts that depict musical instruments can be used to identify the different types of instruments that were used, as well as the way that they were played. This information can then be used to reconstruct medieval music, and to gain a better understanding of the role of music in medieval society. We show two examples of the musiconis library in 8.3 where the first one is a stone sculpture ¹ and the second is a stone painting ² to show case the great diversity of the medieval musical representation.

8.3 AI-powered search

Musicologists rely heavily on museums' websites to find manuscripts related to their field of work, which is a task that has been simplified greatly over the past decade thanks to the digitalization efforts across the European Union. However, such great efforts focused a lot on getting the images online and annotating them with keywords describing the manuscript's context and not the manuscript's content. This makes it very hard for someone to search for a manuscript that contains

¹<https://musiconis.huma-num.fr/fr/fiche/1768/musiciens-jouant-de-la-viele-en-huit-de-l-organistrum-du-frestel-du-monocorde-du-psalterion-de-la-rote-de-la-viele-de-la-harpe-carillon-tintinnabulum-et-acrobate.html>

²<https://musiconis.huma-num.fr/fr/fiche/18/musicien-jouant-de-la-viele-et-tomberesse.html>



Figure 8.1: A stone romane sculpture of a Fiddle-playing musician from the 11th century from Saint-Martin-de-Boscherville, Normandie, France.



Figure 8.2: A stone romane painting of a Fiddle-playing musician from the 13th century from Retjons, Nouvelle-Aquitaine, France.

Figure 8.3: Two example images from Musiconis library of medieval musical artworks.

for example, a Violin, and even harder if the search is on a topic that is more vague and requires complex pattern recognition and visual indicator detection.

AI can play a significant role in this, as it is able to perform simple but repetitive tasks that require basic human intelligence at a much faster pace than an ordinary human being. While working on this thesis, we collaborated with a researcher and musicologist Valerie le Page who has spent many years searching for specific images [8.4](#)³ to help her understand medieval singing practices and help reconstruct the Notre Dame de Paris. AI could have helped her perform her search in a much shorter time.

The idea behind AI-powered search, is that we leverage the images we have already found of the objects we want and then annotate them and train a machine learning model (object detection) on our dataset. The task then becomes to use the model and parse through all the images of manuscripts for each folio and find potential examples of the same objects in other locations. Figure [8.5](#) shows the exact steps to take to perform the search on a museum dataset.

- **Data:** The quality of the results is dependent directly on the quality and quantity of the data used for training. This task usually takes a lot of time to perform as it requires the formulation of a research question, the definition

³BnF ms. fr. 13091 f. 177r Psautier_de_Jean_de_Berry



Figure 8.4: An example of singing in churches used to train the medieval signing search engine.

of the objects to look for, any visual clues that the model can use, other data sources of objects that are similar, potential objects that are similar and can confuse the model to include in the dataset and much more.

- **Model:** Using an improved algorithm to provide better results even in the case of low data quantity and quality. We recommend using DDTL algorithm as it is the most generic and easy to recreate even without machine learning engineering experience as all it needs it to run the code on a colab notebook which only requires a few clicks and zero programming skills. In cases where improved results are required, domain adaptation methods provide much better results than transfer learning but it requires excellent programming and machine learning engineering skills. This step usually takes around 1-2 weeks to tune the perfect model. The weights for the models we used throughout the thesis are maintained in [imadbekkouch/medieval_music_yolov8](https://huggingface.co/imadbekkouch/medieval_music_yolov8) https://huggingface.co/imadbekkouch/medieval_music_yolov8.
- **Search:** This task is the most time-consuming and can take up to months on a single museum library. It takes a lot of time to pass through the millions of images available and requires at least a week of experimentation to find the perfect inference threshold for your model/dataset combination. Usually, the tuning is simple, experiment with a two-day run and see how many false positives you get versus True positives, and as long as the true positives number is acceptable and the total of false positive + true positive is still within humanly verifiable numbers then that is your perfect threshold. if you want to increase the true positives, you will have to drop the threshold, and if you want to decrease your False positives to maintain the ease of validation by humans, then increase the threshold.

These steps are followed by an expert validation of the false positives and true positives to search if there is anything useful in the results. It is important to note that we cannot validate the model based on concepts like mAP, f1-score, or even recall which is the most important metric for us because it is impossible to know how many False negatives the model is making on the dataset as this task will require a human to pass through the dataset manually.

8.4 Results and shortcomings

Although having a mAP number to evaluate the potential of a model mathematically is helpful, it still doesn't tell us whether the method is indeed required. Our AI model has suspected 847 images to contain musical representations, most of these images were sadly wrong, as the model misclassifies typical objects found in

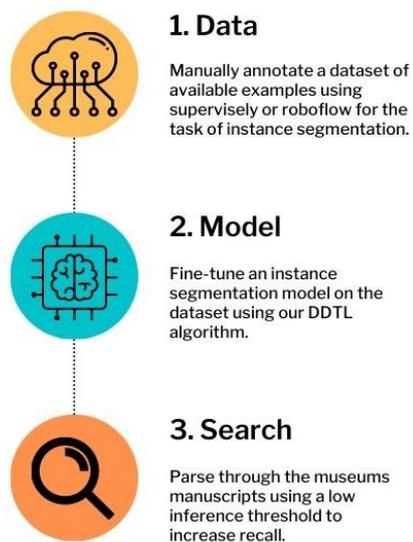


Figure 8.5: Object Detection Search Engine steps.

medieval manuscripts like swords, papers, and shields as musical representations of books and musical instruments. But luckily not all of these images were False Positives; some of them contained exactly what we needed. The model's results were even better than what we realistically expected finding 76 books, 8 lutrins, 113 musical instruments, and 24 singing situations. Although not all books were indeed a representation of singing as 24 is definitely less than 76 but these 24 images can be a big source of information for musicologists and historians alike.

So we have seen that using AI and computer vision to help search in cultural heritage scenarios is useful but is it the only way? To answer such a question we decided to compare the model's search abilities to a text-based search engine and see whether the images we collected are impossible to find otherwise. Using the meta-data and the textual description that is attached to the manuscripts on the museum's website, we tried to see if we could find these exact images through a simple keyword-based search. The majority of the metadata surrounding the manuscripts didn't contain any reference to the musical aspects of it. Only 9 of these manuscripts contained a reference to books and from that only one of these books was indeed a singing situation. The situation gets even worse for musical instruments where the only reference to them was in one manuscript related to King David playing the harp, which is indeed a widespread representation but not what we are looking for in our research, unlike our model, which was able to find 113 different instruments. This clearly shows the importance of using such computer vision techniques to help rebuild our understanding of history and guide our search in the millions of digitized manuscripts out on the internet.

Figure 8.8⁴ Shows two examples of the results of the models that we trained for the tasks of medieval singing and medieval musical instrument recognition.

8.5 Continuous Improvement

"Garbage in, garbage out."

– **Charles Babbage**, English mathematician and inventor (1791-1871)

A quote that encapsulates the most important principle in machine learning and computing in general, if your model is trained on bad data, do not expect good results. This is why it is important to focus on data quality checks and data quantity.

⁴Initial D: Herod Ordering the Massacre of the Innocents; Initial V: Clerics Singing; Unknown; about 1300; Tempera colors, gold leaf, and ink on parchment; Leaf: 26.4 × 18.3 cm (10 3/8 × 7 3/16 in.); Ms. Ludwig IX 3 (83.ML.99), fol. 85v; No Copyright - United States (<http://rightsstatements.org/vocab/NoC-US/1.0/>) The J.P. Getty Museum Ms. Ludwig IX 3 (83.ML.99), fol. 85v



Figure 8.6: Model's results for medieval singing, the manuscript includes a book and lutrin and two singers with traditional religious clothes .



Figure 8.7: Model's results for instrument search.

Figure 8.8: Two example images found by the instance segmentation search engine.

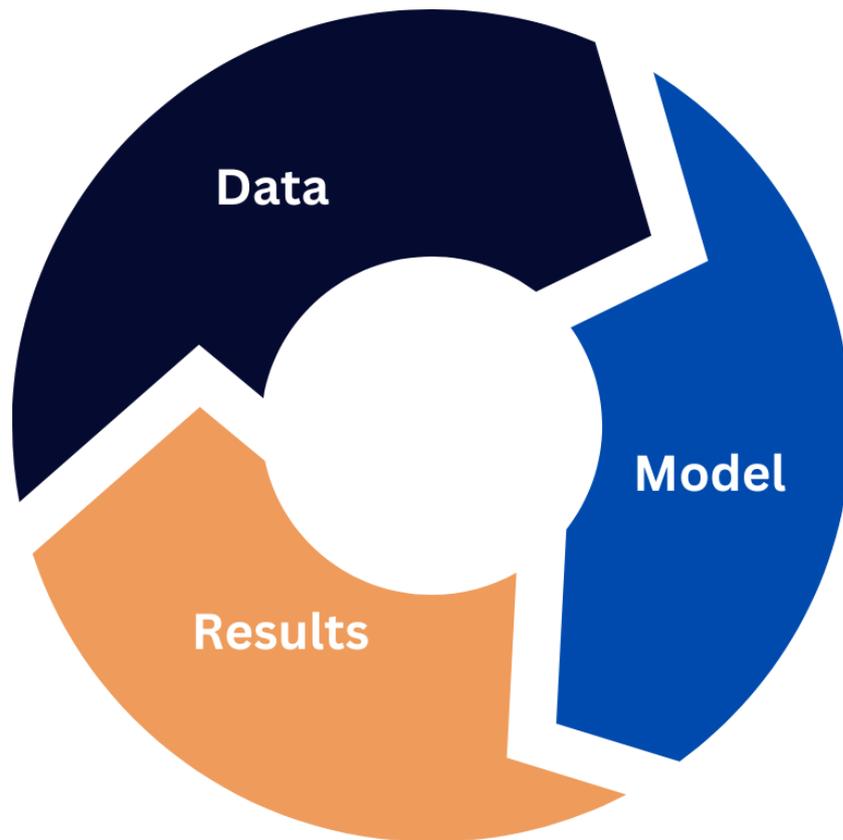


Figure 8.9: A simplified overview of the machine learning pipeline.

In this thesis, we presented many datasets all annotated and validated on multiple passes by musicology experts. Although our results are on par with the current state of the art, they are still far away from human expert-level performance. But, there is a glimpse of hope that we need to remember, the diagram shown in figure 8.9 can be interpreted in two ways. The first is Rubbish in, rubbish out (RIRO) meaning if your data is bad, your model is going to be bad and your results even worse. But it can also signify the cyclic property of this pipeline, you start with bad data and bad models which give you a few results, and then these results will become your data and improve it a little bit. This leads to an improvement of your model which in return leads to more results and so on and so forth.

Getting the wheel started and making that first pass of data, model, and results, will help future researchers go much faster with their passes and get better results. Our contribution to the Musiconis Library with 113 medieval instruments that weren't indexed there before, shows that the work done in this thesis although quite abstract, theoretical, and research-oriented, still has direct implications on practical issues that musicologists face every day.

Chapter 9

Conclusion

In this thesis, we have proposed several methods for improving the current state of the art computer vision models for medieval manuscript studies. Our goal was to experiment with different deep learning and artificial intelligence techniques to help musicologists and human experts in the field of medieval manuscript studies and more specifically historical music. We summarize in the rest of this chapter the different types of contributions we made to the field from new benchmarks and datasets to novel techniques and algorithms for domain adaptation, transfer learning and more. We will also discuss the impact that our research had on the field of medieval manuscript studies and some perspectives for future works.

9.1 Contributions

9.1.1 Datasets and benchmarks

This thesis is in the intersection of artificial intelligence and medieval manuscript studies. Although the AI part of the thesis is a mainstream research topic and is considered a heavily documented field, the medieval manuscript part is still young. One of the main issues we faced during this thesis was to the lack of annotated datasets and hence with the collaboration of a team of 15 musicology students and 4 musicology researchers and experts we managed to annotate multiple large datasets for musical instrument recognition. The selection criteria was very diverse, we started with real photographs of ancient instruments that are stored in museums all around Europe, and then we included historical manuscript artistic depictions of different types of instruments. The main three datasets we annotated are:

1. Annotated Musical Instrument Museums Online (AMIMO): The AMIMO dataset is a High dimensional photographs collection of ancient musical instruments that are stored and maintained in European museums. The goal

of annotating such large dataset (10258 images of different objects) is to allow computer vision models the ability to learn and grasp these musical instruments in their real form. The dataset contains 6 classes which are: Vielles, Lutes, Zithers, Harps, Bows, Lyres.

2. Medieval Musicological Studies Dataset (MMSD): MMSD is a dataset that was manually collected and annotated by musicologists in the goal of understanding the history of medieval signing. The goal is to contribute using the insights found in these images to the reconstruction of the burned Notre-Dame de Paris Cathedral. The dataset is a collection of images of illuminations representing medieval singing in different forms. We annotated 341 images of 5 classes: Phylactery, Folio, Book, Altar, and Lectern.
3. Musical Instruments Recognition In Medieval Artworks: We created this dataset to allow as a collection from two major sources which are Musiconis and Vihuelas:
 - (a) AnnMusiconis Iconography representations for string instruments: The AnnMusiconis database is collection of images that our team at IReMus have created over years of manual search and many collaborations on the french level and even European level. It is a catalog of iconographic representations for music and sound performances since the start of the Middle Ages period until its end. the dataset contains 662 images for string instruments which are: Zithers, Harps, Lutes, Lyres, Vielles.
 - (b) AnnVihuela: The AnnVihuelas database is a Spanish Renaissance musical instruments collection. It contains images from 1470 to 1630. We annotated 165 images with 4 classes which are: Lutes, Vielles, Harps, and Lyres.

Our datasets will be publicly available on the Musiconis website after the musicologists who annotated the datasets publish their thesis on the topics.

9.1.2 Transfer learning techniques

The first step for algorithmic contribution in this thesis went for transfer learning. Transfer learning is a very active field of research in artificial intelligence that leverages multiple one or multiple sources of data to enhance performances on another more challenging set of data. Our contributions can be summarized in two methods, one is for general transfer learning and the second is for the few shot transfer learning.

1. Dual Training for Transfer Learning: is a non-intrusive approach that aims at leveraging the current state of the art object detection models which are

pre-trained on irrelevant but large datasets that allows the models to learn basic concepts like shapes. It improves the performances of such models by incorporating an extra dataset that is similar in style to new photographs yet contains the classes that we are interested. The algorithm is a two step process. The first step aims at focusing the weight update of the object detector on reducing the source domain loss and slowly shifts the interest towards the target domain in the second step. We control the interest of the model using an automatically updated weighting function.

2. Few Shot Object Detection: is a simple non-intrusive method for few shot object detection that integrates seamlessly with the state of the art object detection models such as: YOLO v4-5-6, Faster RCNNs, and even attention based models. The idea aims at first improving the object proposal part of the object detection model regardless of the class, to be able to leverage the acceptable amount of objects we have in different classes without worrying about the small object-per-class issue. The second step of our model aims at training the full model together focusing mainly on the object classification now instead of the object detection part of the loss.

Both methods are black box techniques and can be used with any object detection model.

9.1.3 Auxiliary learning techniques

Getting a computer vision to improve its results on a dataset without adding any extra data is a dream that can only be accomplished using large servers and weeks of fine tuning. The other method is called auxiliary learning, whereby we add more data in the form of meta data or a different way of using the same data to perform an extra task. Our contributions follow the name of the thesis into two sub parts, Knowledge Graph Embeddings auxiliary learning and Adversarial auxiliary learning.

1. Knowledge Graph Embeddings: Our images come with a large metadata set describing the context of The manuscript. we represent this metadata in the knowledge graph we use to build concept embeddings using the node2vec algorithm. Knowledge graph embeddings form a continuous lower dimensional space where concepts and their relations are preserved. Our technique leverages these embeddings to guide the domain adaptation process by combining them with the visual embeddings from the images as anchors that smooth the latent space using an extension of Fisher's linear discriminant.
2. Adversarial Learning: After performing a deep analysis of what makes neural networks overfit, we found a common problem where are the neural network is

extracting too much noise and irrelevant information from the image related to the style and surroundings of the objects and not the objects themselves. Hence we proposed two new solutions that allows the neural network to learn category informative features and at the same time domain and style independent.

- (a) Reconstruction adversarial learning: After performing a deep analysis of what makes neural networks overfit, we found a common problem where are the neural network is extracting too much noise and irrelevant information from the image related to the style and surroundings of the objects and not the objects themselves. Hence we proposed two new solutions that allows the neural network to learn category informative features and at the same time domain and style independent.
- (b) Augmentation adversarial learning : is a simple and fast domain adaptation technique that is based on the same idea of defining a behaviour correlated with over-fitting and domain dependency, building a network that aims for detecting such behavior in the latent space, and training your encoder on the adversarial loss to that. For the sake of simplicity we decided to make the least amount of expectations, by making the model predict the augmentation transformations applied to the input image and the encoder will try to hide and ignore these details in the extraction process which in turn leads to a more generalisable latent space extraction.

9.1.4 Augmenting Musiconis

Validating models when working on small datasets is a very tricky situation. Even when using traditional measures like F1-score, recall, precision and accuracy, there is still a large chance that the results you are getting are due to hyper-parameter tuning or to just luck. During this thesis we aimed at using multiple datasets and benchmarks even outside of the field of medieval manuscripts and reported the average results of at least 5 repeats of the same experiment, but that is not enough. This is why we took it to the next level, by actually using our models on the real data from the BNF dataset where we launched a script that downloads the images of manuscript pages directly from the BNF without applying any pre-processing such as zooming on the drawings or segmenting the image into different parts. The model performed extremely well, providing 165 new manuscripts that were validated by musicologists that contain more than 300 musicology references (signing with religious books, or without) and instruments. The mode's results were very far from perfect as it made many False Positives. Some of the issues the model has made is predicting swords and spears that are aimed at a person's

head/mouth as a wind musical instrument (mainly trumpets) and same also for the shields as a stringed musical instrument (mainly Lutes). The model has suspected a 643 manuscript images and the musicologists have chosen 165 of those images as containing the objects we are looking for. The manuscripts are currently being described and processed to be added to the musiconis dataset to further aid with musicology research.

9.2 Perspectives for future work

The deep learning and artificial intelligence field, in general, is evergrowing and always on continuous improvements, and as we discussed throughout the thesis, our results provide excellent value to musicologists, but they are far from perfection. We list down here a couple of future works to enhance the quality of our results.

9.2.1 Rules and clues

For the majority of the research aims of our musicology colleagues, the target of their interest is associated with a set of rules and guidelines on when is the image acceptable or not. We take the example of Vieles vs. Lutes, since they look similar in shape in an almost eroded sculpture, the only clues they have are the position of the second hand or the presence of arche.

$$\text{shape}(Viele/Lute) \wedge \text{close}(\text{other_hand}) \implies Lute$$

$$\text{shape}(Viele/Lute) \wedge \text{close}(\text{arche}) \implies Viele$$

In other cases we have the presence of a book on a lectern in front of a group of at least three men wearing specific clothes implies the presence of religious singing and rituals. Such examples of rules and clues are everywhere embedded at the core of the musicology research, because finding any book in a manuscript was a very easy task for our models which found many of them but the majority of the 800+ images of books found by the model were useless and only 12 ended up containing the musical performance in question. Building a model and a method that uses these rules and clues as a strength point instead of a failure will provide researchers with exactly what they are looking for in a much faster time.

9.2.2 Leveraging high dimensional data

The majority of the images we worked with are extremely high dimensional with image dimensions up to 7000+ pixels in each direction. All of this information is wasted immediately before our model even see the images. Since we are using



Figure 9.1: DALL-E mini by crayon.com.



Figure 9.2: DALL-E by OpenAI.

Figure 9.3: Two example images generated by AI for the following sentence: "King David playing the harp in medieval manuscript"

pre-trained models such as YOLO v5 and other large models pre-trained on large datasets because we don't have enough data to retrain a large model that can take such images as direct input from scratch. This leads to having the area of interest so small it can be barely visible by the human eye since most of them cover less than 1% of the manuscripts area. A logical solution should train two different systems, one aimed at producing areas of interest followed by another model that zooms into only those areas of interest and performs the operation again on a more zoomed in image, this should allow for way less False Negatives which is the most important measure of error for musicologists.

9.2.3 Image generation

Data scarcity is our biggest problem when working with medieval data. Recent advances in the field of image generation have revolutionized the field of computer vision and natural language processing by bridging the gap between them completely. The main two models currently in use are DALL-E 2 and Google's Imagen. DALL-E is an artificial intelligence system which has been designed by OpenAI that can create hyper realistic images and art from either a description in natural human language or a source image that can be changed slightly. DALL-E was able to learn the relationships between words and their shapes and colors, through a process called "diffusion" which takes a random noise in the shape of an image and continuously alters and makes changes to it towards the target image that



Figure 9.4: Viele in a medieval manuscript.



Figure 9.5: King playing the harp in medieval manuscript.

Figure 9.6: Two example images generated by AI for the following sentence: "King David playing the harp in medieval manuscript"

contains the specific aspects it is looking for.

Figure 9.3 shows a direct application of DALL-E to our problems. It doesn't require an expert to see that even though the model generates hyper-realistic images and art that is better than any human can, it still suffers greatly with such difficult scenarios. Although the model usually generated hyper realistic images for descriptions that are related to different artistic periods and realistic scenarios. As the models by design accept different images as input we can aim to retrain or fine-tune such models to generate new images of objects of interest in more challenging scenarios.

9.2.4 Interpretability

"Why does the model make a mistake?" a question that many researchers in the field of artificial intelligence work on and many musicologists kept asking me throughout the thesis. This is a clear proof that interpretability and reasoning is part of the human nature. Many of the errors of the models that we discussed earlier were in repeating scenarios such as use of weapons close to the mouth or holding similar objects to instruments in a suspicious manner. Building a system that can identify such scenarios where the model lacks in performance can allow the researchers to understand what more data to annotate or even better what scenarios can we ask AI to generate in order to help our object detection models to better understand the underlying research questions of the musicologists.

Bibliography

- [1] T. Adel and A. Wong. *A Probabilistic Covariate Shift Assumption for Domain Adaptation*. 2015.
- [2] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, and M. Atri. “An evaluation of retinanet on indoor object detection for blind and visually impaired persons assistance navigation”. In: *Neural Processing Letters* (2020), pp. 1–15.
- [3] K. Akuzawa, Y. Iwasawa, and Y. Matsuo. “Adversarial Invariant Feature Learning with Accuracy Constraint for Domain Generalization”. In: *CoRR* abs/1904.12543 (2019). arXiv: [1904.12543](https://arxiv.org/abs/1904.12543).
- [4] J. Armitage, E. Kacupaj, G. Tahmasebzadeh, M. Maleshkova, R. Ewerth, and J. Lehmann. “MLM: A Benchmark Dataset for Multitask Learning with Multiple Languages and Modalities”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2967–2974.
- [5] N. Arvanitopoulos, G. Chevassus, D. Maggetti, and S. Süssstrunk. “A handwritten French dataset for word spotting: CFRAMUZ”. In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*. 2017, pp. 25–30.
- [6] B. K. Barakat, J. El-Sana, and I. Rabaev. “The Pinkas Dataset”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2019, pp. 732–737.
- [7] E. Batanina, I. E. I. Bekkouch, Y. Youssry, A. Khan, A. M. Khattak, and M. Bortnikov. “Domain Adaptation for Car Accident Detection in Videos”. In: *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2019, pp. 1–6.
- [8] I. E. I. Bekkouch, D. C. Nicolae, A. Khan, S. M. A. Kazmi, A. M. Khattak, and B. Ibragimov. “Adversarial Reconstruction Loss for Domain Generalization”. In: *IEEE Access* 9 (2021), pp. 42424–42437.

- [9] I. Bekkouch, N. D. Constantin, V. Eyharabide, and F. Billiet. “Adversarial Domain Adaptation for Medieval Instrument Recognition”. In: Manuscript accepted for publication in Proceedings of SAI Intelligent Systems Conference. 2021.
- [10] I. Bekkouch, V. Eyharabide, and F. Billiet. “Dual Training for Transfer Learning: Application on Medieval Studies”. In: Manuscript accepted for publication in Int. Joint Conference on Neural Network. 2021.
- [11] I. Bekkouch, Y. Youssry, R. Gafarov, A. Khan, and A. M. Khattak. “Triplet Loss Network for Unsupervised Domain Adaptation”. In: *Algorithms* 12.5 (May 2019), p. 96.
- [12] I. E. I. Bekkouch, T. Aidinovich, T. Vrtovec, R. Kuleev, and B. Ibragimov. “Multi-agent shape models for hip landmark detection in MR scans”. In: *Medical Imaging 2021: Image Processing*. Vol. 11596. SPIE, 2021, pp. 153–162.
- [13] I. E. I. Bekkouch, N. D. Constantin, V. Eyharabide, and F. Billiet. “Adversarial Domain Adaptation for Medieval Instrument Recognition”. In: *Intelligent Systems and Applications*. Ed. by K. Arai. Cham: Springer International Publishing, 2022, pp. 674–687.
- [14] I. E. I. Bekkouch, V. Eyharabide, and F. Billiet. “Dual Training for Transfer Learning: Application on Medieval Studies”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8.
- [15] I. E. I. Bekkouch, V. Eyharabide, and F. Billiet. “Dual Training for Transfer Learning: Application on Medieval Studies”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8.
- [16] P. Bharati and A. Pramanik. “Deep learning techniques—R-CNN to mask R-CNN: a survey”. In: *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019* (2020), pp. 657–668.
- [17] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. “Learning Bounds for Domain Adaptation”. In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis. Curran Associates, Inc., 2008, pp. 129–136.
- [18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *arXiv preprint arXiv:2004.10934* (2020).
- [19] M. Boillet, M.-L. Bonhomme, D. Stutzmann, and C. Kermorvant. “HO-RAE: an annotated dataset of books of hours”. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. 2019, pp. 7–12.

- [20] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. “Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks”. In: *CoRR* abs/1612.05424 (2016). arXiv: [1612.05424](#).
- [21] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. “Domain Separation Networks”. In: *CoRR* abs/1608.06019 (2016). arXiv: [1608.06019](#).
- [22] G. Cai, Y. Wang, M. Zhou, and L. He. “Unsupervised Domain Adaptation with Adversarial Residual Transform Networks”. In: *CoRR* abs/1804.09578 (2018). arXiv: [1804.09578](#).
- [23] F. M. Carlucci, A. D’Innocente, S. Bucci, B. Caputo, and T. Tommasi. “Domain generalization by solving jigsaw puzzles”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2229–2238.
- [24] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò. “Just DIAL: DomaIn Alignment Layers for Unsupervised Domain Adaptation”. In: *CoRR* abs/1702.06332 (2017). arXiv: [1702.06332](#).
- [25] G. Carneiro, N. P. Da Silva, A. Del Bue, and J. P. Costeira. “Artistic image classification: An analysis on the printart database”. In: *European conference on computer vision*. Springer. 2012, pp. 143–157.
- [26] G. Castellano and G. Vessio. “Deep convolutional embedding for digitized painting clustering”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 2708–2715.
- [27] E. Cetinic. “Iconographic image captioning for artworks”. In: *arXiv preprint arXiv:2102.03942* (2021).
- [28] M. Chen, K. Q. Weinberger, and J. C. Blitzer. “Co-training for Domain Adaptation”. In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS’11. Granada, Spain: Curran Associates Inc., 2011, pp. 2456–2464.
- [29] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang. “A Closer Look at Few-shot Classification”. In: *CoRR* abs/1904.04232 (2019). arXiv: [1904.04232](#).
- [30] V. Cheng and C. Li. “Personalized spam filtering with semi-supervised classifier ensemble”. In: *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI’06)*. IEEE. 2006, pp. 195–201.

- [31] F. Chiabrando, E. Donadio, and F. Rinaudo. “SfM for orthophoto to generation: A winning approach for cultural heritage knowledge”. In: *The international archives of the photogrammetry, remote sensing and spatial information sciences* 40 (2015), pp. 91–98.
- [32] H.-T. Choi, H.-J. Lee, H. Kang, S. Yu, and H.-H. Park. “SSD-EMB: An Improved SSD Using Enhanced Feature Map Block for Object Detection”. In: *Sensors* 21.8 (Apr. 2021), p. 2842.
- [33] N. Cilia, C. De Stefano, F. Fontanella, C. Marrocco, M. Molinara, and A. S. Di Freca. “An end-to-end deep learning system for medieval writer identification”. In: *Pattern Recognition Letters* 129 (2020), pp. 137–143.
- [34] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. “Generative adversarial networks: An overview”. In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.
- [35] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.
- [36] H. Daumé III, A. Kumar, and A. Saha. “Frustratingly easy semi-supervised domain adaptation”. In: *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*. 2010, pp. 53–59.
- [37] J. S. Denker et al. “Neural Network Recognizer for Hand-Written Zip Code Digits”. In: *Advances in Neural Information Processing Systems 1*. Ed. by D. S. Touretzky. Morgan-Kaufmann, 1989, pp. 323–331.
- [38] A. A. Deshmukh, A. Bansal, and A. Rastogi. “Domain2Vec: Deep Domain Generalization”. In: *CoRR* abs/1807.02919 (2018). arXiv: [1807.02919](https://arxiv.org/abs/1807.02919).
- [39] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker. “Domain generalization via model-agnostic learning of semantic features”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 6450–6461.
- [40] A. Elgammal, Y. Kang, and M. Den Leeuw. “Picasso, matisse, or a fake? Automated analysis of drawings at the stroke level for attribution and authentication”. In: *Thirty-second AAAI conference on artificial intelligence*. 2018.
- [41] V. Eyharabide, I. E. I. Bekkouch, and N. D. Constantin. “Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition”. In: *Computers* 10.8 (2021).
- [42] V. Eyharabide, I. E. I. Bekkouch, and N. D. Constantin. “Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition”. In: *Computers* 10.8 (2021), p. 94.

- [43] V. Eyharabide, V. Lully, and F. Morel. “MusicKG: Representations of Sound and Music in the Middle Ages as Linked Open Data”. In: *Semantic Systems*. Ed. by M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, and Y. Sure-Vetter. 2019, pp. 57–63.
- [44] M. F. Ferreira, R. Camacho, and L. F. Teixeira. “Autoencoders as Weight Initialization of Deep Classification Networks for Cancer versus Cancer Studies”. In: *arXiv preprint arXiv:2001.05253* (2020).
- [45] M. Feurer and F. Hutter. “Hyperparameter optimization”. In: *Automated machine learning*. Springer, Cham, 2019, pp. 3–33.
- [46] C. Finn, P. Abbeel, and S. Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135.
- [47] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. “A survey on concept drift adaptation”. In: *ACM computing surveys (CSUR)* 46.4 (2014), pp. 1–37.
- [48] Y. Ganin and V. Lempitsky. “Unsupervised domain adaptation by back-propagation”. In: *arXiv preprint arXiv:1409.7495* (2014).
- [49] Y. Ganin et al. “Domain-Adversarial Training of Neural Networks”. In: *Journal of Machine Learning Research* 17.59 (2016), pp. 1–35.
- [50] Y. Ganin et al. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.
- [51] N. Garcia et al. “A dataset and baselines for visual question answering on art”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 92–108.
- [52] G. A. Gesese, R. Biswas, M. Alam, and H. Sack. “A survey on knowledge graph embeddings with literals: Which model links better literal-ly?” In: *Semantic Web Preprint* (2019), pp. 1–31.
- [53] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. “Domain Generalization for Object Recognition with Multi-task Autoencoders”. In: *CoRR* abs/1508.07680 (2015). arXiv: [1508.07680](https://arxiv.org/abs/1508.07680).
- [54] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. “Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation”. In: *CoRR* abs/1607.03516 (2016). arXiv: [1607.03516](https://arxiv.org/abs/1607.03516).
- [55] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. *Detectron*. <https://github.com/facebookresearch/detectron>. 2018.
- [56] R. B. Girshick. “Fast R-CNN”. In: *CoRR* abs/1504.08083 (2015). arXiv: [1504.08083](https://arxiv.org/abs/1504.08083).

- [57] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *CoRR* abs/1311.2524 (2013). arXiv: [1311.2524](https://arxiv.org/abs/1311.2524).
- [58] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. “Google vizier: A service for black-box optimization”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 1487–1495.
- [59] N. Gonthier, Y. Gousseau, S. Ladjal, and O. Bonfait. “Weakly supervised object detection in artworks”. In: *European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.
- [60] A. Hamid, M. Bibi, M. Moetesum, and I. Siddiqi. “Deep Learning Based Approach for Historical Manuscript Dating”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2019, pp. 967–972.
- [61] D. M. Hawkins. “The problem of overfitting”. In: *Journal of chemical information and computer sciences* 44.1 (2004), pp. 1–12.
- [62] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X.-S. Hua. “

\

alpha –IoU : A family of power intersection over union losses for bounding box regression”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20230–20242.

- [63] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV) (2017)*, pp. 2980–2988.
- [64] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. “Mask R-CNN”. In: *CoRR* abs/1703.06870 (2017). arXiv: [1703.06870](https://arxiv.org/abs/1703.06870).
- [65] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385).
- [66] K. He, X. Zhang, S. Ren, and J. Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.
- [67] L. Hu, M. Kan, S. Shan, and X. Chen. “Duplex Generative Adversarial Network for Unsupervised Domain Adaptation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

- [68] L. Hu, M. Kan, S. Shan, and X. Chen. “Duplex generative adversarial network for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1498–1507.
- [69] P. Hu, M. Xu, M. Wu, G. Chen, and C. Zhang. “Handwritten Style Recognition for Chinese Characters on HCL2020 Dataset”. In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer. 2020, pp. 138–150.
- [70] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf. “Correcting Sample Selection Bias by Unlabeled Data”. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS’06. Canada: MIT Press, 2006, pp. 601–608.
- [71] B. I. Ibrahim, D. C. Nicolae, A. Khan, S. I. Ali, and A. Khattak. “VAE-GAN Based Zero-Shot Outlier Detection”. In: *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control*. ISCSIC 2020. Newcastle upon Tyne, United Kingdom: Association for Computing Machinery, 2020.
- [72] B. I. E. Ibrahim, V. Eyharabide, V. Le Page, and F. Billiet. “Few-Shot Object Detection: Application to Medieval Musicological Studies”. In: *Journal of Imaging* 8.2 (2022).
- [73] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167 (2015). arXiv: [1502.03167](https://arxiv.org/abs/1502.03167).
- [74] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. “A survey on contrastive self-supervised learning”. In: *Technologies* 9.1 (2020), p. 2.
- [75] R. Janković. “Machine learning models for cultural heritage image classification: Comparison based on attribute selection”. In: *Information* 11.1 (2019), p. 12.
- [76] G. Jocher et al. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. Version v3.1. Oct. 2020.
- [77] D. Kadish, S. Risi, and A. S. Løvlie. “Improving Object Detection in Art Images Using Only Style Transfer”. In: *arXiv preprint arXiv:2102.06529* (2021).
- [78] M. Kan, S. Shan, and X. Chen. “Bi-Shifting Auto-Encoder for Unsupervised Domain Adaptation”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 3846–3854.

- [79] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. “Few-shot object detection via feature reweighting”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8420–8429.
- [80] G. Kang, J. Li, and D. Tao. “Shakeout: A new approach to regularized deep neural network training”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.5 (2017), pp. 1245–1258.
- [81] M. Kaselimi, N. Doulamis, A. Doulamis, A. Voulodimos, and E. Protopadakis. “Bayesian-optimized Bidirectional LSTM Regression Model for Non-intrusive Load Monitoring”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 2747–2751.
- [82] J. Kiesel, F. Kneist, L. Meyer, K. Komlossy, B. Stein, and M. Potthast. “Web Page Segmentation Revisited: Evaluation Framework and Dataset”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 3047–3054.
- [83] B. Kiessling, D. S. B. Ezra, and M. T. Miller. “Badam: A public dataset for baseline detection in arabic-script manuscripts”. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. 2019, pp. 13–18.
- [84] M. Krause. *machine learning - What is translation invariance in computer vision and convolutional neural network? - Cross Validated*. <https://stats.stackexchange.com/questions/208936/what-is-translation-invariance-in-computer-vision-and-convolutional-neural-netwo>. (Accessed on 04/22/2022).
- [85] D. Krompaß, S. Baier, and V. Tresp. “Type-constrained representation learning in knowledge graphs”. In: *The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I 14*. Springer. 2015, pp. 640–655.
- [86] K. Lambers et al. “Learning to look at LiDAR: The use of R-CNN in the automated detection of archaeological objects in LiDAR data from the Netherlands”. In: *Journal of Computer Applications in Archaeology* 2.1 (2019), pp. 31–40.
- [87] L. Le, A. Patterson, and M. White. “Supervised autoencoders: Improving generalization performance with unsupervised regularizers”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 107–117.

- [88] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.
- [89] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski. “A theoretical framework for back-propagation”. In: *Proceedings of the 1988 connectionist models summer school*. Vol. 1. San Mateo, CA, USA. 1988, pp. 21–28.
- [90] B. C. G. Lee et al. “The Newspaper Navigator Dataset: Extracting Headlines and Visual Content from 16 Million Historic Newspaper Pages in Chronicling America”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 3055–3062.
- [91] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. “Deeper, Broader and Artier Domain Generalization”. In: *CoRR* abs/1710.03077 (2017). arXiv: [1710.03077](https://arxiv.org/abs/1710.03077).
- [92] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot. “Domain generalization with adversarial feature learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5400–5409.
- [93] K. Li, Z. Huang, Y.-C. Cheng, and C.-H. Lee. “A maximal figure-of-merit learning approach to maximizing mean average precision with deep neural network based classifiers”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 4503–4507.
- [94] Z. Li, B. Ko, and H. Choi. “Pseudo-Labeling Using Gaussian Process for Semi-Supervised Deep Learning”. In: *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. Jan. 2018, pp. 263–269.
- [95] Z. Li, L. Qu, Q. Xu, and M. Johnson. “Unsupervised pre-training with Seq2Seq reconstruction loss for deep relation extraction models”. In: *Proceedings of the Australasian Language Technology Association Workshop 2016*. 2016, pp. 54–64.
- [96] Z. Lipton, Y.-X. Wang, and A. Smola. “Detecting and Correcting for Label Shift with Black Box Predictors”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3122–3130.
- [97] L. Liu et al. “Deep learning for generic object detection: A survey”. In: *International journal of computer vision* 128 (2020), pp. 261–318.
- [98] M. Liu, T. Breuel, and J. Kautz. “Unsupervised Image-to-Image Translation Networks”. In: *CoRR* abs/1703.00848 (2017). arXiv: [1703.00848](https://arxiv.org/abs/1703.00848).

- [99] M.-Y. Liu, T. Breuel, and J. Kautz. “Unsupervised image-to-image translation networks”. In: *Advances in neural information processing systems*. 2017, pp. 700–708.
- [100] M. Liu and O. Tuzel. “Coupled Generative Adversarial Networks”. In: *CoRR* abs/1606.07536 (2016). arXiv: [1606.07536](https://arxiv.org/abs/1606.07536).
- [101] S. Liu, A. Davison, and E. Johns. “Self-supervised generalisation with meta auxiliary learning”. In: *Advances in Neural Information Processing Systems* 32 (2019).
- [102] Z. Liu et al. “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 10012–10022.
- [103] M. Long, J. Wang, and M. I. Jordan. “Deep Transfer Learning with Joint Adaptation Networks”. In: *CoRR* abs/1605.06636 (2016). arXiv: [1605.06636](https://arxiv.org/abs/1605.06636).
- [104] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao. “Local geometric structure feature for dimensionality reduction of hyperspectral imagery”. In: *Remote Sensing* 9.8 (2017), p. 790.
- [105] F. Lv, J. Zhu, G. Yang, and L. Duan. “TarGAN: Generating target data with class labels for unsupervised domain adaptation”. In: *Knowledge-Based Systems* 172 (2019), pp. 123–129.
- [106] L. van der Maaten and G. E. Hinton. “Visualizing Data using t-SNE”. In: 2008.
- [107] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci. “Best sources forward: domain generalization through source-specific nets”. In: *CoRR* abs/1806.05810 (2018). arXiv: [1806.05810](https://arxiv.org/abs/1806.05810).
- [108] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci. “Boosting Domain Adaptation by Discovering Latent Domains”. In: *CoRR* abs/1805.01386 (2018). arXiv: [1805.01386](https://arxiv.org/abs/1805.01386).
- [109] M. Mehri, P. Héroux, R. Mullot, J.-P. Moreux, B. Coüasnon, and B. Barrett. “HBA 1.0: A pixel-based annotated dataset for historical book analysis”. In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*. 2017, pp. 107–112.
- [110] F. Milani and P. Fraternali. “A Dataset and a Convolutional Model for Iconography Classification in Paintings”. In: *Journal on Computing and Cultural Heritage (JOCCH)* 14.4 (2021), pp. 1–18.

- [111] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. “Unified deep supervised domain adaptation and generalization”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5715–5725.
- [112] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. “Unified Deep Supervised Domain Adaptation and Generalization”. In: *CoRR* abs/1709.10190 (2017). arXiv: [1709.10190](https://arxiv.org/abs/1709.10190).
- [113] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: 2011.
- [114] A. Y. Ng. “Feature selection, L 1 vs. L 2 regularization, and rotational invariance”. In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 78.
- [115] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. “A review of relational machine learning for knowledge graphs”. In: *Proceedings of the IEEE* 104.1 (2015), pp. 11–33.
- [116] S. J. Pan and Q. Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (Oct. 2010), pp. 1345–1359.
- [117] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. “Transferrable Prototypical Networks for Unsupervised Domain Adaptation”. In: *CoRR* abs/1904.11227 (2019). arXiv: [1904.11227](https://arxiv.org/abs/1904.11227).
- [118] P. O. Pinheiro. “Unsupervised Domain Adaptation with Similarity Learning”. In: *CoRR* abs/1711.08995 (2017). arXiv: [1711.08995](https://arxiv.org/abs/1711.08995).
- [119] V. Pondenkandath, M. Alberti, N. Eichenberger, R. Ingold, and M. Liwicki. “Cross-Depicted Historical Motif Categorization and Retrieval with Deep Learning”. In: *Journal of Imaging* 6.7 (2020), p. 71.
- [120] I. Rabaev, B. K. Barakat, A. Churkin, and J. El-Sana. “The HHD Dataset”. In: *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE. 2020, pp. 228–233.
- [121] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015). arXiv: [1506.02640](https://arxiv.org/abs/1506.02640).
- [122] J. Redmon and A. Farhadi. “YOLO9000: Better, Faster, Stronger”. In: *CoRR* abs/1612.08242 (2016). arXiv: [1612.08242](https://arxiv.org/abs/1612.08242).
- [123] J. Redmon and A. Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: [1804.02767](https://arxiv.org/abs/1804.02767).

- [124] R. Remus. “Domain Adaptation Using Domain Similarity- and Domain Complexity-Based Instance Selection for Cross-Domain Sentiment Analysis”. In: *2012 IEEE 12th International Conference on Data Mining Workshops*. Dec. 2012, pp. 717–723.
- [125] S. Ren, K. He, R. B. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: [1506.01497](https://arxiv.org/abs/1506.01497).
- [126] Z. Ren and Y. J. Lee. “Cross-Domain Self-Supervised Multi-task Feature Learning Using Synthetic Imagery”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 762–771.
- [127] M. T. Ribeiro, S. Singh, and C. Guestrin. “" Why should I trust you?" Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.
- [128] M. T. Ribeiro, S. Singh, and C. Guestrin. *Model-Agnostic Interpretability of Machine Learning*. 2016. arXiv: [1606.05386](https://arxiv.org/abs/1606.05386) [stat.ML].
- [129] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. “Contractive Auto-Encoders: Explicit Invariance During Feature Extraction”. In: *ICML*. 2011.
- [130] A. R. Rivera, A. Khan, I. E. I. Bekkouch, and T. S. Sheikh. “Anomaly Detection Based on Zero-Shot Outlier Synthesis and Hierarchical Feature Distillation”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2020), pp. 1–11.
- [131] M. ROBERTS et al. “The Application of Machine Learning to At-Risk Cultural Heritage Image Data”. PhD thesis. Durham University, 2020.
- [132] O. Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.
- [133] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. “From source to target and back: symmetric bi-directional adaptive GAN”. In: *CoRR* abs/1705.08824 (2017). arXiv: [1705.08824](https://arxiv.org/abs/1705.08824).
- [134] M. Sabatelli, M. Kestemont, W. Daelemans, and P. Geurts. “Deep transfer learning for art classification problems”. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

- [135] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. “Adapting Visual Category Models to New Domains”. In: *Computer Vision – ECCV 2010*. Ed. by K. Daniilidis, P. Maragos, and N. Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 213–226.
- [136] K. Saito, Y. Ushiku, and T. Harada. “Asymmetric Tri-training for Un-supervised Domain Adaptation”. In: *CoRR* abs/1702.08400 (2017). arXiv: [1702.08400](https://arxiv.org/abs/1702.08400).
- [137] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. “Maximum classifier discrepancy for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3723–3732.
- [138] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. “Generate to adapt: Aligning domains using generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8503–8512.
- [139] O. Sener, H. O. Song, A. Saxena, and S. Savarese. “Learning Transferrable Representations for Unsupervised Domain Adaptation”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 2110–2118.
- [140] P. Sermanet, S. Chintala, and Y. LeCun. “Convolutional Neural Networks Applied to House Numbers Digit Classification”. In: *CoRR* abs/1204.3968 (2012). arXiv: [1204.3968](https://arxiv.org/abs/1204.3968).
- [141] M. Seuret, S. Limbach, N. Weichselbaumer, A. Maier, and V. Christlein. “Dataset of Pages from Early Printed Books with Multiple Font Groups”. In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. 2019, pp. 1–6.
- [142] S. Sharafi, S. Fouladvand, I. Simpson, and J. A. B. Alvarez. “Application of pattern recognition in detection of buried archaeological sites based on analysing environmental variables, Khorramabad Plain, West Iran”. In: *Journal of Archaeological Science: Reports* 8 (2016), pp. 206–215.
- [143] Z. Shen, K. Zhang, and M. Dell. “A Large Dataset of Historical Japanese Documents with Complex Layouts”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 548–549.
- [144] S. Sheng and M.-F. Moens. “Generating captions for images of ancient artworks”. In: *ACM International Conference on Multimedia*. 2019, pp. 2478–2486.

- [145] G. Shi, H. Huang, and L. Wang. “Unsupervised Dimensionality Reduction for Hyperspectral Imagery via Local Geometric Structure Feature Learning”. In: *IEEE Geoscience and Remote Sensing Letters* 17.8 (2020), pp. 1425–1429.
- [146] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni. “U-net and its variants for medical image segmentation: A review of theory and applications”. In: *Ieee Access* 9 (2021), pp. 82031–82057.
- [147] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold. “Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts”. In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE. 2016, pp. 471–476.
- [148] V. Souza, D. M. d. Reis, A. G. Maletzke, and G. E. Batista. “Challenges in Benchmarking Stream Learning Algorithms with Real-world Data”. In: *arXiv preprint arXiv:2005.00113* (2020).
- [149] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.
- [150] G. Strezoski and M. Worring. “Omniart: multi-task deep learning for artistic data analysis”. In: *arXiv preprint arXiv:1708.00684* (2017).
- [151] C. Szegedy et al. “Going Deeper With Convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.
- [152] Y. Tang et al. “Visual and Semantic Knowledge Transfer for Large Scale Semi-Supervised Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.12 (2018), pp. 3045–3058.
- [153] Y. Teng, A. Choromanska, and M. Bojarski. “Invertible Autoencoder for domain adaptation”. In: *arXiv preprint arXiv:1802.06869* (2018).
- [154] A. Torralba and A. A. Efros. “Unbiased look at dataset bias”. In: *CVPR 2011*. 2011, pp. 1521–1528.
- [155] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou. “Training data-efficient image transformers amp; distillation through attention”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10347–10357.

- [156] A. Tsymbal. “The problem of concept drift: definitions and related work”. In: *Computer Science Department, Trinity College Dublin 106.2* (2004), p. 58.
- [157] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. “Adversarial Discriminative Domain Adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [158] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. “Adversarial Discriminative Domain Adaptation”. In: *CoRR* abs/1702.05464 (2017). arXiv: [1702.05464](https://arxiv.org/abs/1702.05464).
- [159] D. Valy, M. Verleysen, S. Chhun, and J.-C. Burie. “A new khmer palm leaf manuscript dataset for document analysis and recognition: Sleukrith set”. In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*. 2017, pp. 1–6.
- [160] J. Vanschoren. “Meta-learning”. In: *Automated Machine Learning*. Springer, Cham, 2019, pp. 35–61.
- [161] A. Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
- [162] A. Wang, H. Hu, and L. Yang. “Image captioning with affective guiding and selective attention”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.3 (2018), pp. 1–15.
- [163] H. Wang, Z. He, Z. C. Lipton, and E. P. Xing. “Learning Robust Representations by Projecting Superficial Statistics Out”. In: *CoRR* abs/1903.06256 (2019). arXiv: [1903.06256](https://arxiv.org/abs/1903.06256).
- [164] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng. “End-to-end object detection with fully convolutional network”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15849–15858.
- [165] Q. Wang, Z. Mao, B. Wang, and L. Guo. “Knowledge Graph Embedding: A Survey of Approaches and Applications”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.
- [166] Y.-Q. Wang. “An analysis of the Viola-Jones face detection algorithm”. In: *Image Processing On Line* 4 (2014), pp. 128–148.
- [167] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu. “Frustratingly Simple Few-Shot Object Detection”. In: (July 2020).
- [168] Y. Wang and Q. Yao. “Few-shot learning: A survey”. In: (2019).

- [169] J. Wei, J. Liang, R. He, and J. Yang. “Learning Discriminative Geodesic Flow Kernel for Unsupervised Domain Adaptation”. In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. July 2018, pp. 1–6.
- [170] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. “Bam! the behance artistic media dataset for recognition beyond photography”. In: *IEEE international conference on computer vision*. 2017, pp. 1202–1211.
- [171] B. Wu et al. “Visual Transformers: Token-based Image Representation and Processing for Computer Vision”. In: *CoRR* abs/2006.03677 (2020). arXiv: [2006.03677](https://arxiv.org/abs/2006.03677).
- [172] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. <https://github.com/facebookresearch/detectron2>, 2019.
- [173] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. “A comprehensive survey on graph neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.
- [174] R. Xie, J. Wen, A. Quitadamo, J. Cheng, and X. Shi. “A deep auto-encoder model for gene expression prediction”. In: *BMC genomics* 18.9 (2017), p. 845.
- [175] S. Xie, Z. Zheng, L. Chen, and C. Chen. “Learning Semantic Representations for Unsupervised Domain Adaptation”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 5423–5432.
- [176] Z. Xu, M. Wilber, C. Fang, A. Hertzmann, and H. Jin. “Learning from multi-domain artistic images for arbitrary style transfer”. In: *arXiv preprint arXiv:1805.09987* (2018).
- [177] E. Yahaghi, A. Movafeghi, S. Ahmadi, S. Ansari, M. Taheri, and N. Rastkhah. “Cultural Heritage Object Identification by Radiography Nondestructive Method and Digital Image Processing”. In: *Applied Mechanics and Materials* 83 (2011), pp. 35–40.
- [178] K. Yakovlev, I. E. I. Bekkouch, A. M. Khan, and A. M. Khattak. “Abstraction-Based Outlier Detection for Image Data”. In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2020, pp. 540–552.
- [179] K. Yakovlev, I. E. I. Bekkouch, A. M. Khan, and A. M. Khattak. “Abstraction-Based Outlier Detection for Image Data”. In: *Intelligent Systems and Applications*. Ed. by K. Arai, S. Kapoor, and R. Bhatia. Cham: Springer International Publishing, 2021, pp. 540–552.

-
- [180] H. Yang, Z. Lin, and M. Zhang. “Rethinking Knowledge Graph Evaluation Under the Open-World Assumption”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 8374–8385.
- [181] T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei. “Semi-supervised Domain Adaptation with Subspace Learning for visual recognition”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 2142–2150.
- [182] H. Zhang, L. Liu, Y. Long, and L. Shao. “Unsupervised Deep Hashing With Pseudo Labels for Scalable Image Retrieval”. In: *IEEE Transactions on Image Processing* 27.4 (Apr. 2018), pp. 1626–1638.
- [183] J. Zhu, T. Park, P. Isola, and A. A. Efros. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *CoRR* abs/1703.10593 (2017). arXiv: [1703.10593](https://arxiv.org/abs/1703.10593).

Auxiliary learning & Adversarial training pour les études des manuscrits médiévaux

Résumé

Cette thèse se situe à l'intersection de la musicologie et de l'intelligence artificielle, et vise à exploiter l'IA pour aider les musicologues dans leur travail répétitif, comme la recherche d'objets dans les manuscrits du musée. Nous avons annoté quatre nouveaux ensembles de données pour l'étude des manuscrits médiévaux : AMIMO, AnnMusiconis, AnnVihuelas et MMSD. Dans la deuxième partie, nous améliorons les performances des détecteurs d'objets en utilisant des techniques de Transfer learning et de Few Shot Object Detection. Dans la troisième partie, nous discutons d'une approche puissante de Domain Adaptation, qui est auxiliary learning, où nous formons le modèle sur la tâche cible et une tâche supplémentaire qui permet une meilleure stabilisation du modèle et réduit le over-fitting. Enfin, nous abordons l'apprentissage auto-supervisé, qui n'utilise pas de méta-données supplémentaires en tirant parti de l'approche de adversarial learning, forçant le modèle à extraire des caractéristiques indépendantes du domaine.

Mots-clés : Apprentissage auxiliaire ; réseaux antagonistes ; graphe de connaissances ; réseaux de neurones ; Patrimoine culturel ; Études des manuscrits médiévaux

Auxiliary learning & Adversarial training for Medieval Manuscript Studies

Summary

This thesis is at the intersection of musicology and artificial intelligence, aiming to leverage AI to help musicologists with repetitive work, such as object searching in the museum's manuscripts. We annotated four new datasets for medieval manuscript studies: AMIMO, AnnMusiconis, AnnVihuelas, and MMSD. In the second part, we improve object detectors' performances using Transfer learning techniques and Few Shot Object Detection. In the third part, we discuss a powerful approach to Domain Adaptation, which is auxiliary learning, where we train the model on the target task and an extra task that allows for better stabilization of the model and reduces over-fitting. Finally, we discuss self-supervised learning, which does not use extra meta-data by leveraging the adversarial learning approach, forcing the model to extract domain-independent features.

Keywords : Auxiliary learning ; adversarial training ; Knowledge Graph Embeddings ; Neural Network Embeddings ; Cultural Heritage ; Medieval Manuscript Studies

UNIVERSITÉ SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE :

ED 5 – Concepts et langages

Maison de la Recherche, 28 rue Serpente, 75006 Paris, FRANCE

DISCIPLINE : Informatique pour les sciences humaines et sociales